

Heidelberg University
Institute of Computer Science
Visual Computing Group (VCG)

Master-Arbeit

Light Transport Techniques for Tensor Field Visualization

Sebastian Bek

Matrikelnummer: 3481802

Betreuer: Sebastian Bek

Datum der Abgabe: 07.07.2019

dt.: Ich versichere, dass ich diese Master-Arbeit selbstständig verfasst und nur die angegebenen Quellen und Hilfsmittel verwendet habe und die Grundsätze und Empfehlungen “Verantwortung in der Wissenschaft” der Universität Heidelberg beachtet wurden.

eng.: I hereby declare/assure, that I drafted this thesis independently and only used the sources and materials labeled as references and that the conventions/principles and recommendations “Verantwortung in der Wissenschaft” of the Heidelberg University have been regarded/observed.

Abgabedatum / Due Date: 07.07.2019

Zusammenfassung

Tensorfelder werden meistens in Verbindung mit mechanischen Spannungsverteilungen in 2D/3D-Gittern gebracht (vgl. Cauchy-Stress Tensor), haben aber auch andere praktische Bedeutungen in der Physik. Mithilfe von globalen Beleuchtungsmodellen/-techniken wird eine neue Methode entwickelt, um Tensorfelder zu visualisieren, die aber zusätzlich auch verwendet werden kann um die Lichtausbreitung in einer topologischen Szene zu beschreiben. Als Grundlage leiten wir ein einfaches Lichtausbreitungsschema für kartesische Gitter ab, dass die Prinzipien der Ausbreitungsdämpfung und Energieerhaltung beachtet und Lichtverteilung/en für gegebene Lichtquellenrichtung/en und -positionen bis zur Konvergenz approximieren kann. Als unterliegendes Modell werden die Transmissionsprofile innerhalb diesem Gitter als Kristallfiberstruktur angenommen (vgl. Edelsteine: Katzenaugeneffekt beim Tigerauge). Die folgende Aufgabe ist es, anisotrope Fiberstrukturen mit der Orientierung und den Anisotropiemaßen des unterliegenden Tensorfelds zu modellieren. Dafür erstellen wir per Hauptachsentransformation (PCA) ein Eigensystem für jede Zelle des Tensorfelds und modellieren eine zugehörige Ellipsengleichung als (Transfer-/Wichtungs-) Transmissionsprofil. Folglich kann die resultierende Lichtverteilung als 2D Skalarfeld oder Polarplots für Richtungsinformation visualisiert/repräsentiert werden. Wir messen Impulsantworten des Tensorfelds mit Delta-Pulsen an jeder Position und in jede Richtung um eine uniform gesamplete Map der globalen Lichtverteilungen zu erhalten. Diese Map wird als ein globales Energieflussfeld betrachtet. Wir lassen uns von der Technik FTLE aus dem Bereich Particle Tracing der Vector Field Visualization inspirieren, indem wir den Gradient des Flussfelds der resultierenden Lichtverteilungen analysieren und visualisieren. Ähnlich wie FTLE in Vektorfeldern Gräte/Kämme/LCS detektiert, erfasst unsere Methode dieselben Strukturen, anziehende/abstoßende/sattelnde Tensorfeldlinien die häufig Schlüsselstrukturen in der Natur repräsentieren, in Tensorfeldern. Wir führen diese Größe als Global Illumination Gradient ein, der einen FTLE-verwandten Ansatz darstellt, um LCS in Tensorfeldern durch die Analyse des globalen, durch einen Imprint (Gravur in kristallinen Fiberstrukturen) gerichteten Lichtflusses, zu visualisieren.

Abstract

Many problems in science and engineering require tensor field representations. Tensor fields, are most commonly associated with stress distributions (cf. Cauchy-stress tensor) in 2D/3D grids, but also have some other meanings in practice in physics. By means of global illumination techniques, we motivate a new method to visualize tensor fields, which is also capable of visualizing the light transport (propagation) in a topologically defined scene geometry. As a basis, we derive a simple light propagation scheme for Cartesian grids, which satisfies propagation attenuation and energy-conservation principles and is able to approximate light distributions for given light source position(s) and direction(s) until convergence. The transmission profiles within this grid are considered as crystal fiber structures as an underlying physical model (, eg., gemstones: tiger’s eye’s cat’s eye effect). The consequent task is to model anisotropic fibre structures with the orientation and anisotropy measures of the underlying tensor field in the grid. For this, we derive an eigenbasis by PCA (principal component analysis) for every cell of the tensor field and form an ellipsoid equation as transmission (transfer/weighting) function for the transmitted light profiles. Hence, the resulting light distribution can be visualized via 2D scalar field color coding or polar plots for directional information. Additionally, we measure impulse responses of the tensor field with Dirac-pulses as light sources at any (sampled) position and in any direction to generate a uniform sampled map of the global illumination distributions. This map is considered as a global illumination energy flow field. We gain inspiration by vector field particle tracing’s FTLE approach for analyzing the gradient of the flow field of the resulting light distributions. Similar to how the FTLE detects ridges/LCS (Lagrangian coherent structures) in vector fields (flow fields), our approach captures these structures likewise, namely attracting/repelling/saddling tensor field lines (TFL) which frequently represent key structures in nature, in tensor fields. We denote this entity a light transport gradient (LTG) which states an FTLE-related approach for visualizing LCS in tensor fields through analysis of the global light transport directed by imprint/engraving in crystalline fiber structures modeled by the tensor field’s ellipsoid glyphs. We evaluate our approaches for plausibility and comparability within an extensive test campaign and demonstrate its performance on several examples of datasets.

Contents

1	Introduction	1
1.1	Motivation	1
1.2	Objectives	2
1.3	Structure of this Work	4
2	Fundamentals and Recent Work	6
2.1	Tensor Fields	6
2.2	Polar Coordinates	8
2.3	Principal Component Analysis	9
2.4	GT data / Approaches	11
2.5	Related Work	13
3	Method	17
3.1	Requirements and Ambitions	17
3.2	Light Transport (Propagation) Scheme	18
3.3	Transmission Profiles and Weighting Functions	22
3.4	Physical Model	26
3.5	Light Transport Gradient (LTG)	27
4	Results and Evaluation	29
4.1	Propagation Attenuation	29
4.2	Light Propagation Scheme Behaviour	29
4.3	Behaviour for Total Anisotropy	32
4.4	Energy Conservation Test	32
4.5	Light transport gradient (LTG)	33
5	Zusammenfassung und Ausblick	37

1 Introduction

In this work, we will discuss the advantages of tensor field analysis and motivate a new method for tensor field visualization through a coherent light transport visualization. At first, we will shortly state the problem setting, objectives and give a quick overview about the structure of this work. The main contributions of this work are listed in the bullet list below:

Contributions of this Work

- light transport method (proagation scheme) following basic but crucial physical principles for tensor field visualization using tensors as transmission profiles
- visualization of the propagation results in $2D$ polar profiles
- a FTLE-related approach called light transport gradient (LTG) for visualizing key structures (ROIs) namely LCS in $2D$ second-order tensor fields

1.1 Motivation

Tensor field visualization is gaining importance as a relevant tool for the analysis of fluid and solid mechanics, where tensor representations are crucially needed. This is always the case, when a directional distribution is needed for a single point in space. Tensors occur most commonly in medical, scientific and engineering applications, making them a frequently used mathematical entity in science. In general, a tensor is a α -D generalization of scalar, vector and matrix and appears, e.g., as Jacobian matrix (spatial gradient) in vector (flow) field visualization or as stress tensor in solid and fluid continuum mechanics. Basically, tensors describe compressions, tensions, rotations and volume changes in both solid and fluid material. While most techniques focus on symmetric tensor field visualization (like glyphs and tensor field lines), some other recent works adress the problem of asymmetric tensor field visualization. The issue with asymmetric (non-symmetric) or antisymmetric (skew-symmetric) tensors is, that they do not always yield real eigenvalues, which is needed to determine an eigenbasis to set up

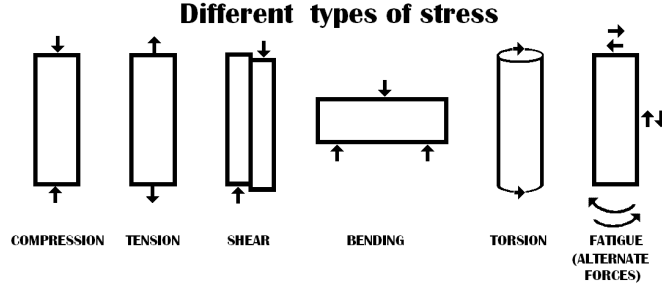


Figure 1.1: Different types of stress

src: https://upload.wikimedia.org/wikipedia/commons/e/eb/DIFFERENT_TYPES_OF_STRESS.png

glyphs, hyperstreamlines or tensor field lines (classically). Nevertheless, there has been extensive work in this domain as well. Zheng & Phang et al. [1] proposed the concept of dual eigenvectors and Lin et al. [2] extended it to pseudo-eigenvectors and introduced eigenvector and eigenvalue manifold to visualize eigenvectors in the complex domain. In this work, we will use the **U**-matrix obtained from singular value decomposition (SVD) to obtain the half-axes of the PCA principal ellipsoid representing the tensor transformation as suggested by Moler [3] to state another alternative of processing asymmetric tensors. 2D tensor field visualization is useful for cases where only two-dimensional data is available, such as satellite remote sensing data, Particle Imagery Velocimetry (PIV) data and earthquake data. Additionally, tensor field visualization aids geologists and medical scientists in their field of work to display complicated number arrays which seem almost impossible to interpret in their raw, unprocessed form. Nevertheless, comprehensive visualization is aiding scientists in their everyday lives understanding complex data loads such as tensor fields. We can only try to invent new methods continuously evaluating them for suitability for certain domains of science to provide our techniques at best considering them for the right applications.

1.2 Objectives

The main purpose of visualization in general is to reduce or encode some set of numbers in some readable, explorable and more intuitive representation. This implies, e.g., clustering, structuring or projection techniques. When it comes to tensor fields, this comes in handy in, e.g., medical sciences such as in DT-MRI (diffusion tensor-medical resonance imaging) [6] when measuring water molecule diffusion characteristics within

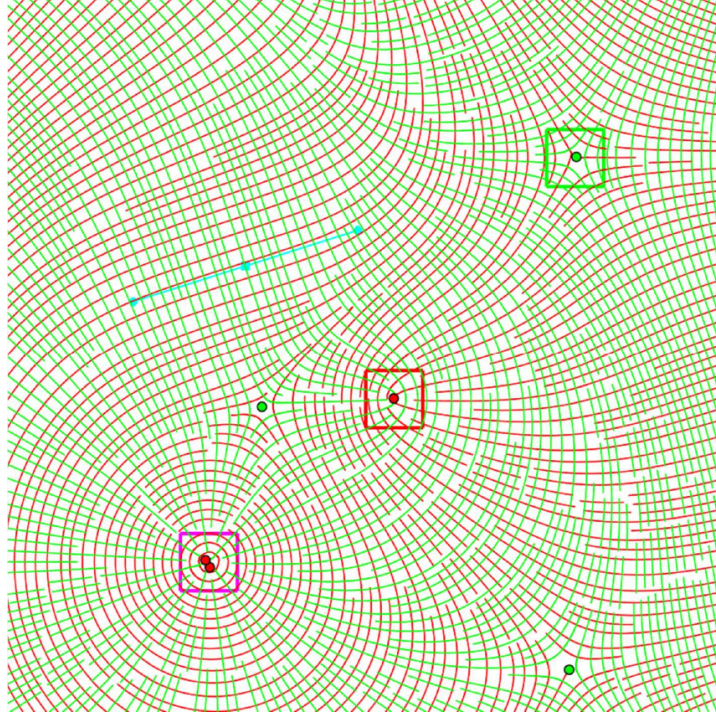


Figure 1.2: Hyperstreamlines in tensor field visualization

src: <http://www2.cs.uh.edu/~chengu/Teaching/Fall2018/Lecs/Lec17.pdf>

tissue or in simulated (or measured) stress tensor fields from solid and fluid mechanics. In these domains of science the problem arises, that tensor fields can be generated (simulated) or measured with physical/mathematical background, but not be regarded intuitively. Necessity begets ingenuity as always, which lets visualization methods and systems emerge in this context. Concurrently, the problem setting of this work encompasses the visualization of tensor fields, i.e., mostly 2D/3D-grids consisting of mainly second-order tensors (cf. Cauchy stress tensor). These grids span a spatial domain in general with sample positions indicated by vertices. Tensor fields feature a sampling grid consistent of tensors which is usually spaced uniformly. A Cartesian grid, which we will utilize in this context, is a normalized regular grid. We aim to use global illumination methods to model light transport directed by tensor fields. Thus, the objective of this work is to design a simple light propagation scheme yet following the most basic but crucial physical principles and laws, like energy conservation and propagation attenuation for tensor field visualization. Additionally, it should hold the possibility to model light transport transport in a geometrical scene. This propagation scheme can either be used to visualize the light transport paths (defined by separatrices) in a scene defined by geometry concerning no influence of a tensor field. Or, on the other hand, be used to visualize the anisotropy characteristics of a given tensor field concerning the influence of

one. However, the tensor field needs to be pre-processed somehow to generate a tensor induced footprint map of transmission profiles, which we will do via matrix decompositions. For this purpose, we will let the transmission profiles act on our simple yet realistic light propagation scheme to have a kind of tensor controlled directive effect on the light transport. The transmission profiles should modulate the outgoing intensities in a sense of redirection or redistribution. Of course, the propagation scheme will need some kind of stimulus which can be defined through symbolic functions by the user which we sample as a primary step. This way, the approach will operate as a global illumination visualization technique for light distributions in polar coordinates but the user must set an initial profile into the grid. In the sense of automatization and optimization, another consequent aim is to implement an approach borrowed from vector-field visualization to reveal structures, which are hard to grasp by ordinary methods. We construct an FTLE-related approach in tensor field visualization (both symmetric and asymmetric) for detecting ridges/LCS (Lagrangian coherent structures). These structures can be classified (identified) as repelling, attracting and saddling (forming a saddle) local features, which frequently represent key structures in nature. LCS usually separate regions of different flow behavior, but in our case they separate regions/domains of varying directional distributions (manifesting in diverging/converging tensor field lines) represented by tensors as we use our tensor controlled light transport method to compute the FTLE field. Thus, we propose another method to compute an FTLE like field on tensor fields. The FTLE map responds most where tensor field lines / hyperstreamlines converge or diverge, whereas one is the same as the other because of bidirectional stress directions and since we effectively measure the finite-time Lyapunov exponent (FTLE) with both forward and backward integration time (stimulus directions, in our case), we will detect both cases. Both invented methods, will give new insights into the topology and anisotropy of tensor fields and provide tools for the structural analysis. The greater goal of this work is therefore the encoding of a tensor field into certain elaborate representations and the interpretation of meaningful results comparing the proposed techniques against others. The designed approach could e.g. be applied in medicinal sciences, engineering and physics to reveal structures which are hard to grasp through interpreting simpler and sometimes confusing glyph and tensor field line visualizations.

1.3 Structure of this Work

For the rest of this work, we will give a quick overview on the structure and outline. Next, in chap. 2, we will capture important fundamentals like Tensors, polar coordinates, PCA and GT data in the context of singular value decomposition (generalization

of eigenvalue decomposition). Also we will discuss the most relevant related work in the domain of tensor field visualization. In chap. 3, we will propose the main contributions and methods of this work and document the results and evaluation in chap. 4. Subsequently, A compact Summary, Conclusion and Outlook will be given in chap. 5.

2 Fundamentals and Recent Work

In this chapter, we will capture the prerequisites necessary to understand the contributions of this work and we will capture tensor fields, polar coordinates, PCA, GT approaches and related work in that sense.

2.1 Tensor Fields

Now what is a tensor? A tensor is a o -dimensional generalization of a matrix as illustrated in Table 2.1. In addition to that, a tensor has as many indices as its order o and their run length is as long as its embedded dimension n in space (the dimension of the row and column vectors), which is equal to the rank of the tensor for full rank (in other words the number of elements n for a $n \times n$ -matrix). Tensors follow certain transformation rules which are defined for covariant/contravariant tensors. They are most commonly obtained and defined from mathematical transformation equations. Hence, any oD number array could be a tensor, but the definition holds only if the transformation rules apply to these oD number arrays (in a sense of mathematical/formal interpretation). This is mostly true for component indexed oD number entities following matrix multiplication and transformation rules found in physics and math manifesting in calculation rules like matrix multiplication. Thus, a tensor in general is a oD number array arranged in meaningful order to put it simple. In fact, tensors themselves are interpreted as transformations which reflect an incoming direction vector (measuring, e.g., the stress in that direction) to an outgoing (stress) vector. To be a bit more concrete, tensors represent stress distributions in solids and fluids describing strain and flow features as these entities are not simple numbers or vectors. They occur in form of a tensor fields, and thus as entities or quantities in physics. That is, at each point (typically a Euclidean space or manifold) in space (indexed by the grid) there is a whole

Table 2.1: Tensor Shapes

order	0	1	2	3	...	o
shape	scalar	vector	matrix	“3D matrix”	...	oD matrix

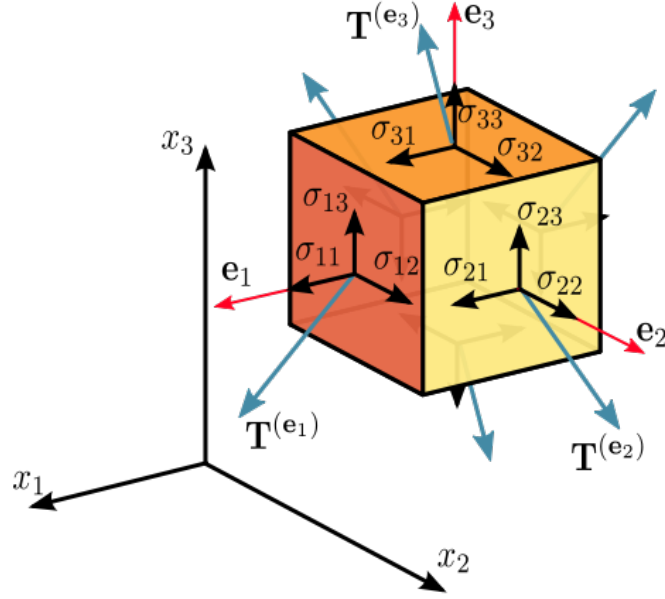


Figure 2.1: Cauchy stress Tensor

src: https://en.wikipedia.org/wiki/Cauchy_stress_tensor#/media/File:Components_stress_tensor_cartesian.svg

distribution of directions, e.g., stresses (elastic and viscous) which needs to be characterized and represented by a single tensor each. A tensor is considered non-degenerate if all eigenvalues $\lambda_i > \lambda_{i+1}$ manifesting in significant anisotropy.

stress tensor The Cauchy stress tensor, which is probably the classical example of a tensor, is depicted in Fig. 2.1 and consists of 3 stress vectors $\mathbf{T}(\mathbf{e}_i)$ arranged in row-major order. It appears in similar form as viscous stress tensor in fluid mechanics:

$$\boldsymbol{\sigma} = \begin{bmatrix} \mathbf{T}(\mathbf{e}_1) \\ \mathbf{T}(\mathbf{e}_2) \\ \mathbf{T}(\mathbf{e}_3) \end{bmatrix} = \begin{bmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{xz} \\ \sigma_{yx} & \sigma_{yy} & \sigma_{yz} \\ \sigma_{zx} & \sigma_{zy} & \sigma_{zz} \end{bmatrix} = \begin{bmatrix} \sigma_x & \tau_{xy} & \tau_{xz} \\ \tau_{yx} & \sigma_y & \tau_{yz} \\ \tau_{zx} & \tau_{zy} & \sigma_z \end{bmatrix} = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ \sigma_{21} & \sigma_{22} & \sigma_{23} \\ \sigma_{31} & \sigma_{32} & \sigma_{33} \end{bmatrix}.$$

These 3 stress vectors represent the orientation and magnitude of total resulting stress at plane x, y, z in direction x, y, z ($3^2 = 9$ individual numbers). Thus, the stress tensor poses a full valid representation of the stress distribution at any point in space. The stress tensor itself can be interpreted as a transformation which maps an incoming normal direction vector \mathbf{n} a resulting stress vector $\mathbf{T}(\mathbf{n}) = \mathbf{n} \cdot \boldsymbol{\sigma}$. The stresses in normal directions (diagonal elements) are directly related to the pressure, which is always isotropic and hence non-directionally dependent (kind of a Euclidean distance of the stress components). The tensor itself is a complicated representation and needs to be projected

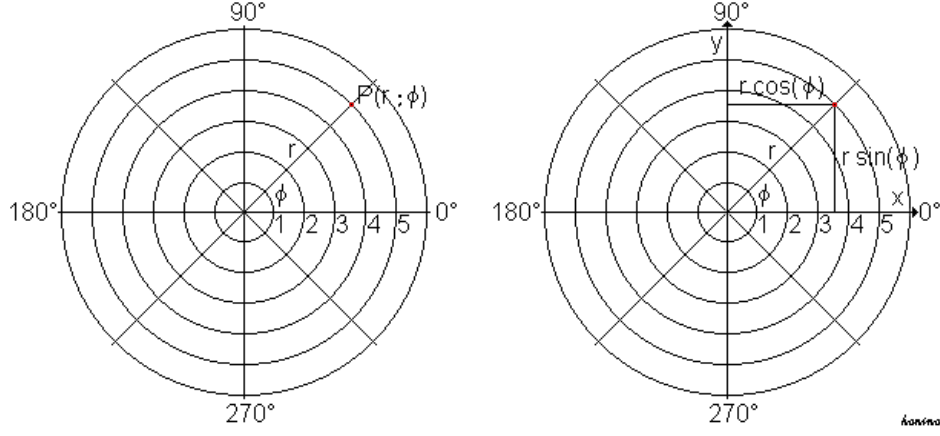


Figure 2.2: polar space

src: https://de.wikipedia.org/wiki/Datei:Ebene_polarkoordinaten.PNG

onto its principal axes to lower its dimensionality. It is possible to find 2 principal stress directions in 2D through PCA (see sect. 2.3 below), for which there are no shear stresses existent in equilibrium. These form the principal directions of deformation at a certain location and are sufficient to describe the resulting transformation/deformation behaviour. The resulting system of principal directions and absolute stresses is often called an “eigensystem” or “eigenframe”. So to say, an eigensystem, consistent of the principal directions/axes, spans a principal ellipsoid depicted in Fig. ?? in sect. 2.3. Another interpretation for the principal component ellipsoid is that it is the result of the tensor transformation applied to the unit circle. It is therefore somehow representing the distortion of an absolutely isotropic object (unit circle) by whatever the affine tensor transformation effects (e.g. stretching, shearing, mirroring, scaling, rotating). Let us consider it as some lower dimensional, more intuitive representation of the tensor which we were left with in the first place. To put it all together, a tensor is a mathematical transformation which is encoded inside of an abstract oD number array, which comes in need, whenever we have a whole distribution of directions manifesting itself in a transformation ellipsoid.

2.2 Polar Coordinates

Mathematically speaking, a polar space is a system which maps each angle ϕ a magnitude $r(\phi)$ through a symbolic function. The conversion formula from polar (given a radius r

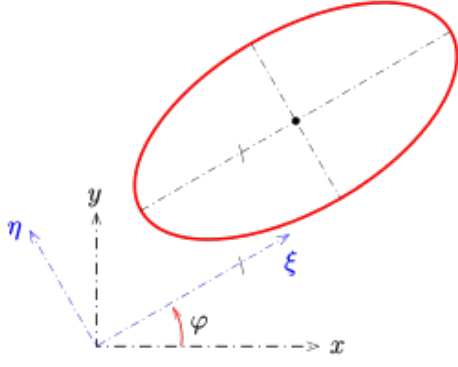


Figure 2.3: 2D principal component ellipsoid glyph (eigenframe)

src: <https://de.wikipedia.org/wiki/Datei:HAT-ellipse0.svg>

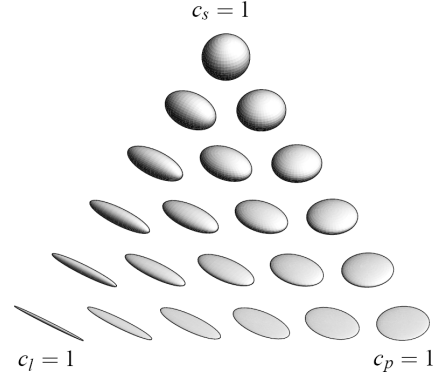


Figure 2.4: 3D ellipsoid glyphs w. anisotropy coefficient (l:linear, p:planar, s:spherical)

src: <https://slideplayer.com/slide/5291764/>

and angle ϕ) to Cartesian coordinates is depicted in Fig. 3.1 and is given by:

$$x = r \cos \phi,$$

$$y = r \sin \phi.$$

The inverse direction (given coordinates x, y) is a bit more complicated:

$$r = \sqrt{x^2 + y^2}$$

$$\phi = \text{atan2}(y, x)$$

The polar domain is a circular axis version of the domain $[0..2\pi]$, with negative magnitudes in r shifted by 180 degrees (point reflected) as this is the formal behaviour defined and required by the conversion formulas.

2.3 Principal Component Analysis

Note that we denote a matrix with a second order tensor interchangeably for the sake of simplicity in the following. Also keep in mind, that we interpret a matrix or tensor as affine transformation. For any state of stress in equilibrium it is possible to find n (dimensionality) independent orthogonal directions with no shear stresses existent. Along these directions the principal stresses, which are normal (tensile/compressive) stresses, are exerted. The anisotropy can be represented by, e.g., ellipsoid glyphs.

Principal Component Analysis is an algorithm which can capture the principal components (directions) in stochastic data. But it works similarly on transformations to capture the principal directions from transformations like rotation and scaling. Now, principal component analysis can be done by eigenvalue or singular vector decomposition, whereas the former yields complex eigenvalues for asymmetric matrices. For (skew-) symmetric (normal) matrices, they are connected through the following relation: $s_i = \sqrt{|\lambda_i|}$ (cf. [4]). Note, that the eigenvectors match the singular vectors only for symmetric matrices, which does not mean that they do not form the half-axes of the principal component ellipsoid in other cases. Mathematically speaking, a decomposition is somehow decomposing a transformation (which follows certain transformation rules) into parts of subsequent transformations consistent of rotation and scaling ($\mathbf{A} = \mathbf{R}\mathbf{S}\mathbf{R}^*$). We choose to use singular value decomposition ($\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^*$) to be able to set up real eigenvalues for any kind of matrix (singular values are always real and positive). Also, the singular vectors of the \mathbf{U} -matrix represent the half-axes of the principal ellipsoid (eigenframe) shown in Fig. ?? for any kind of transformation matrix as pointed out by Moler [3]. The singular value decomposition is defined as follows:

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}$$

where

- \mathbf{U} is an $m \times m$ unitary matrix
- $\mathbf{\Sigma}$ is a diagonal matrix $m \times n$ with positive real numbers on the diagonal
- \mathbf{V} is an $n \times n$ unitary matrix, \mathbf{V}^* is its conjugate transpose

In addition, we also calculate the eigenvalues to exploit the sign of the real part of the eigenvalues ([5]) ordered decreasingly by absolute value (corresponding to singular value order). The sign corresponds to tensile (+) and compressive (−) stress, which would allow us to visualize deformation glyphs with arrows pointing inwards for compressive and outwards for tensile stresses. We do this decomposition analysis once for each matrix and then store the precomputed results in a grid. The principal ellipsoid is also used as a transmission profile for the propagation scheme in symbolic form in a subsequent step, which makes it a basic and central concept within the scope of this work. In this work, we will use the singular vectors yielded from singular value decomposition to avoid complex values yielded from eigenvalue decomposition. This will allow us to set up glyphs representing the major/minor axis of the underlying transformation for every kind of (asymmetric/symmetric) tensor. We will then use global-illumination techniques

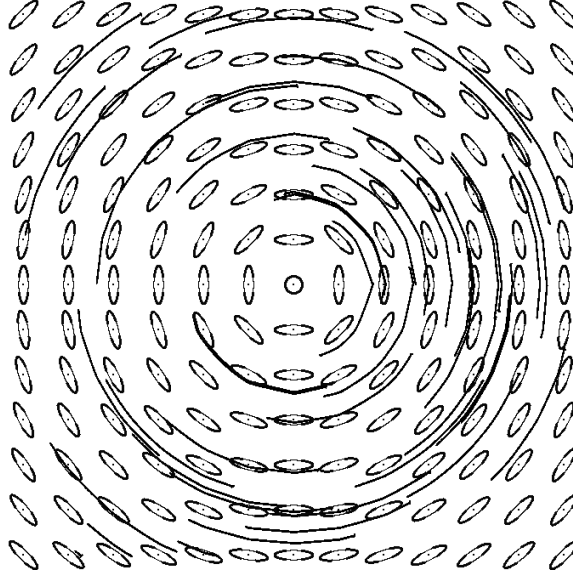


Figure 2.5: tensor field lines: bold lines; ellipsoids: ellipsoid glyphs

on top to generate a light transport flow map, which permits us somehow to forward the intensities which are given as pre-defined initialization values in form of symbolic functions defined by the user. This kind of light transport propagation scheme will also allow us to compute an FTLE-like field on the tensor fields.

2.4 GT data / Approaches

Glyphs and tensor field lines Glyphs are derived as a basis for the weighting profiles of the tensors in the following propagation scheme. They were chosen because it is an easy-to-use (understand) concept which can also be interpreted nicely. In fact, these reps are used for the principal component analysis of direction vectors for the tensor field line extrapolation (integration) in a subsequent step. Tensor Field Lines (the type of tensor stream line - hyperstreamline analog that does not introduce artificial inertia) were implemented for Ground Truth (GT) data generation purposes. For example, when we design an FTLE-like field computation approach, we will need to know when tensor field lines diverge/converge.

Test Data (Generation) We used python's numpy library to generate pre-defined test cases and caught one real data example from [TensorVis.org](https://www.tensorvis.org), which needed to be pre-processed through slicing and subsequent cropping to 2D samples (2×2) matrices), which can be grouped into symmetric and asymmetric ones:

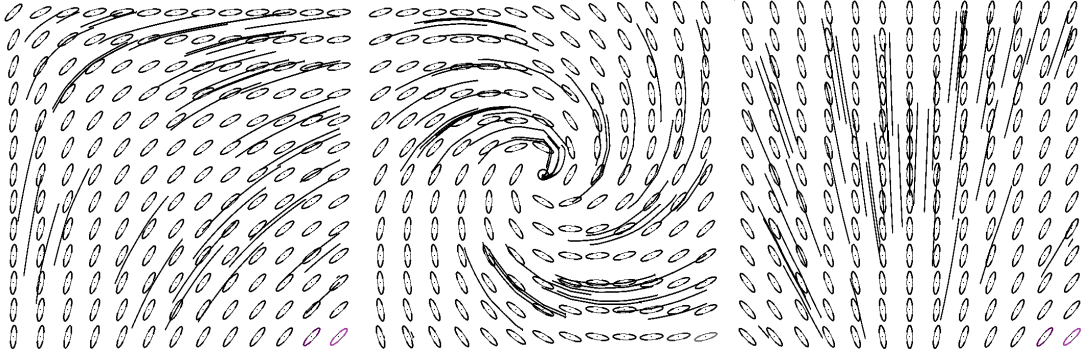


Figure 2.6: a) “corner”-Testfield, b) “spiral”-Tesfield, c) “V”-Tesfield

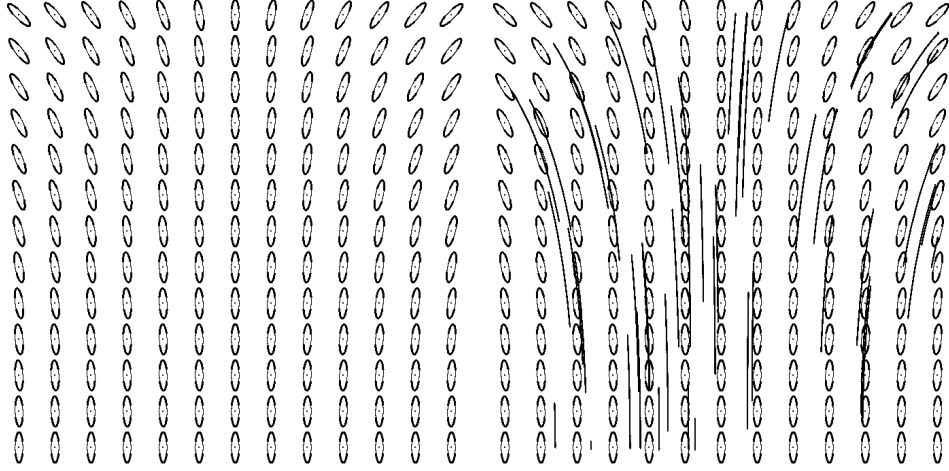


Figure 2.7: a) “drain”-testfield, b) tensor field lines for a)

Symmteric Tensor Fields

Asymmetric Tensor Fields

The “drain”-Testfield, depicted in Fig. 2.7 is generated to have a kind of proof-of-concept (POC) for the FTLE-related approach. It poses raw, bare diverging tensor field lines without any other feature modulated on top. Thus, it can be expected to induce a high FTLE response, since it is an interpretable ground truth test and is considered to be a basic trigger/stimulus for the approach.

The “rings” test field, depicted in Fig. 2.8 should be used to demonstrate the directive ability of the approach transmitting energy in circular orbits. The “spiral” test field, depicted in Fig. 2.9 should be used to demonstrate the directive ability of the approach transmitting energy in archimedian spiral orbits. With this test field we want to demonstrate the directive separation of intensities into four quadrants.

The test field “corner” poses a sustained turn over the grid representing the redirection

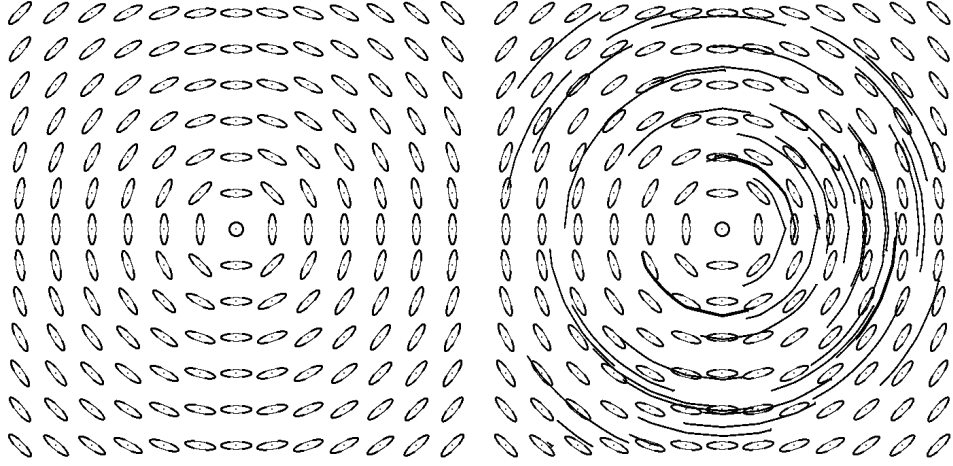


Figure 2.8: a) “rings” test field, b) tensor field lines for a)

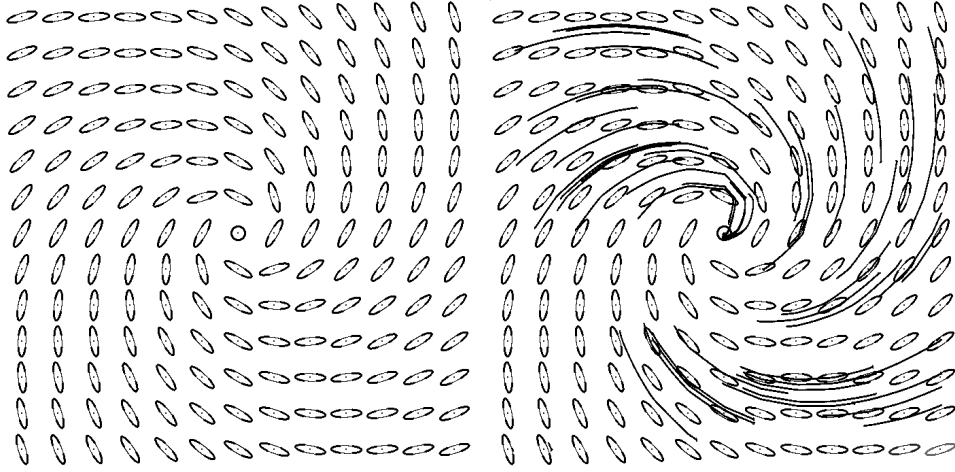


Figure 2.9: a) “spiral”-testfield, b) tensor field lines for a)

or forwarding along domain corners.

2.5 Related Work

There has been extensive recent work on symmetric tensors but in comparison little on asymmetric ones. Since we aren’t able to cover the whole scope in this work, we will focus on the most relevant works on tensor field visualization. We will talk about the publications from Hlawatsch et al. [6] and Zirr et al. [7] in particular, since they pose thematically close related works.

Symmetric Tensor Field Visualization

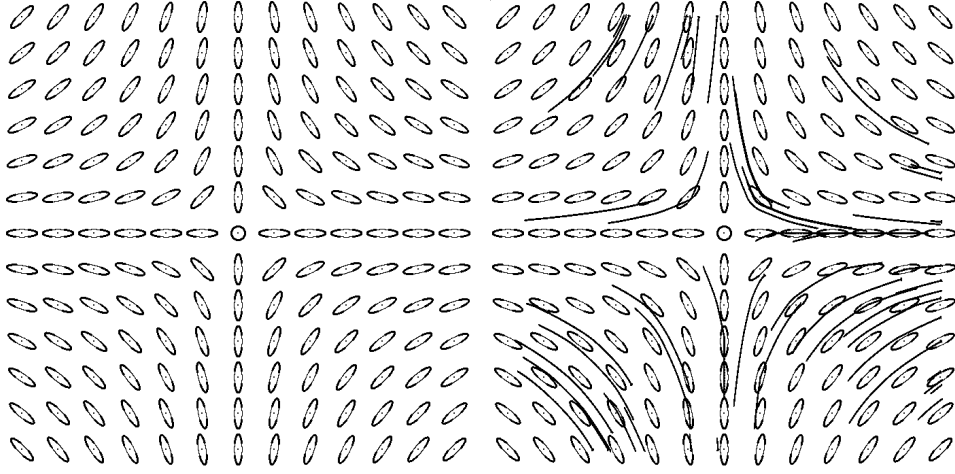


Figure 2.10: a) “inverse”-testfield, b) tensor field lines for a)

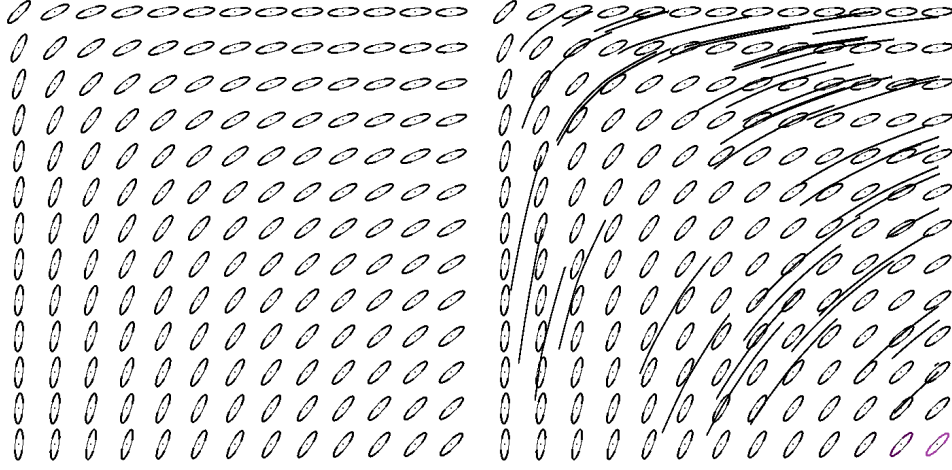


Figure 2.11: a) “corner”-testfield, b) tensor field lines for a)

Zheng & Pang [1] et al. proposed a texture-based visualization approach called Hyper LIC, which extends the concept of LIC to symmetric tensor fields by using an anisotropic 2D-filter kernel oriented along the major/minor eigenvectors. The concept of following the major/minor eigenvector along tensor field lines was invented by Vilanova et al. [8]. Tensorlines [9] introduce a kind of artificial inertia (running average), making them resistant to noise. The notion of tensor field topology (critical, degenerate points, separatrices) and the concept of Hyperstreamlines was first introduced by Delmarcelle and Hesselink [10]. The topology in 3D tensor fields was first analyzed by [11]. The placement of tensor field lines as hyperstreamlines was recently improved by Spencer et al. [12] for glyph packing. A good overview of seeding strategies concerning glyphs is given by McLoughlin et al. [13]. Feng et al. [14] used Voronoi Tesselation for placing

the glyphs. Superquadric glyphs have been proposed by Schultz and Kindlmann [15] for general (non-positive definite) tensors. De Leeuw and van Wijk [16] visualize the partial derivative gradient, the Jacobian of the tensor field to sense the local properties of the field. A diffusion tensor can also be visualized by box [17], ellipsoid [18], composite [19] or superquadric [20] glyphs. There is also a set of scalar measures available in symmetric tensor field visualization such as: mean diffusivity, fractional anisotropy [21] and anisotropy coefficients [22]. These coefficients can also be visualized by direct volume rendering [23]. Hlawatsch et al. [6] obtain a coordinate (flow) map from tensor field lines following major/minor eigenvectors (fibre trajectories) and use the maximum eigenvalue of the Cauchy-Green deformation tensor to compute the FTLE-field in homogeneous regions. In a similar manner, we aim to use light transport techniques to obtain a final light distribution through successive light propagation cycles footprinted (in form of transmission profiles obtained from ellipsoid glyphs) by the tensor field. DT-MRI diffusion tensor imaging can help to reveal the functional [24] or topological [25] structure of the brain and muscles by analysing cerebral and muscle tissue tractography which yields a tensor field of the anisotropy of the brain's white matter. An overview about DT-MRI visualization techniques is given by [8]. Tricoche et al. [26] use invariant manifolds to extract topological features from tensor fields. Other, feature extraction based approaches extract general features as: edges [27], tensor shape orientation [28], interfaces [29], creases [30].

Asymmetric Tensor Field Visualization

Zheng and Pang [1] proposed the concept of dual eigenvectors and Zhang et al. [2] extended it to pseudo-eigenvectors and introduced the eigenvector/eigenvalue manifolds to visualize eigenvectors in the complex domain. Laramée et al. [31] focused on the efficient implementation and visualization of these structures and provided an interactive visualization system for asymmetric tensors applicable in fluid and solid dynamics. The concept of tensor magnitude has been introduced by [31] for means of physical interpretation. They also proposed an efficient glyph and hyperstreamline hybrid approach, which made dynamic interaction in real-time in 2D tensor fields feasible. Palacios et al. [32] extracted isosurfaces of tensor magnitude, mode and isotropy index.

Last, we will discuss our work in relation to other publications. While these previously mentioned works use classical approaches like tensor field lines and glyphs, we will also introduce the concept of Lagrangian coherent structures into our framework contribution of tensor field analysis. LCS have been defined as time-dependent analog of separatrices [33] which are concerned to be robust under noisy conditions [34]. The work of [6] et al. obtains similar results (an FTLE-like field) but obtains them

by sampling of fiber trajectories instead of global illumination light transport methods, as in our case. Our work shares the same basic idea, deriving and computing a FTLE field for tensor field visualization generalizing FTLE from vectors to tensors, yet using a light propagation scheme and hence use whole distributions of trajectories employing polar profiles. We will detect LCS with an FTLE-derived method considering final light transport distributions. The FTLE-related approach named “light transport gradient” to be designed states a generalization of the light transport visualization method FTPD proposed by Zirr et al. [7], since it allows a similar measure to be computed through defining a geometrical scene and setting the transmission (tensor field) to an isotropic and constant value of 100%. In this operating mode, the approach operates as a light transport visualization technique neglecting any tensor field transmission profiles.

3 Method

This chapter will summarize the main contributions of this work studied and state targeted aims and goals as a first step in sect. ???. Then we will have a short introduction to the implemented global illumination scheme derived and simplified from an nVidia global illumination approach in sect. 3.2. Last, we will state the effectively implemented Visualization methods light transport visualization (propagation scheme) and light transport gradient proposed as the main innovatively invented tensor visualization techniques.

3.1 Requirements and Ambitions

Our ambition is to motivate a simple and efficient global illumination propagation scheme able to distribute energy profiles in discrete polar form in 2D/3D-space w.r.t. energy conservation and propagation attenuation principles. We then generate transmission profiles from the eigensystems (principal component frames) of the tensors for each cell in the domain (field). This step is done, to provide an orientation for a grid of crystal fibre structures (cf. optical fibres) to redirect the intensities in analogy to the anisotropies of the tensor field within a glyph representation, obtained from Principal Component Analysis. Our approach should be implicitly adaptive to any kind of 2D dataset or slices of 3D datasets. Thus, we require it to be compatible to handle any kind of matrix (symm./anti-symm.) and any kind of resolution of the field (fully-adaptive). At last, our ambition is to segment key locations (LCS/ridges) in the field with tensor field lines (hyperstreamlines) converging/diverging, which is necessarily the same (for bidirectional eigenvectors/singular vectors of tensors). Thus, we aim to generate a new FTLE-related method for tensor field visualization by applying a light transport gradient to the resulting light distributions for every possible position and direction of the light source in the grid generated by stimulation with Dirac-pulses. Our approach should efficiently handle (relatively) large datasets with reasonable resolutions, which turns out to be possible with additional computing power empowered by a cluster.

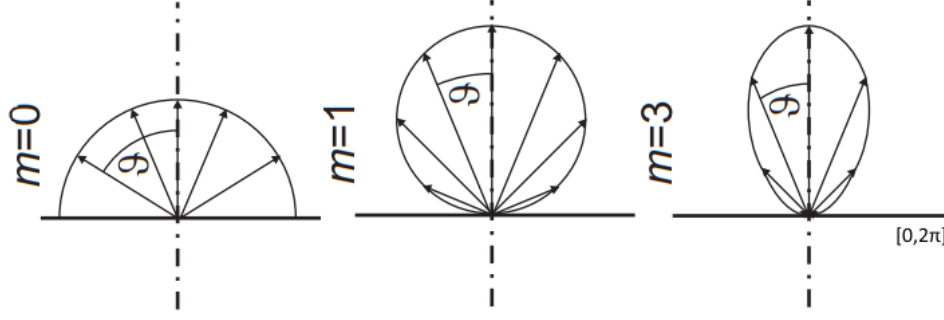


Figure 3.1: Polar profiles for different types of emitters ($m = 0$: spherical emitter, $m = 1$: Lambert emitter, $m = 2$: cone emitter)

src: http://www.optik.uni-erlangen.de/odem/vorlesung/ws1213/Photometrie_Geometrische_Optik.pdf

3.2 Light Transport (Propagation) Scheme

As already mentioned, we will motivate a light transport propagation scheme designed to efficiently propagate initial intensity profiles stored in polar coordinates given as symbolic impulses (stimuli) by the user or the config. This polar profiles are first sampled into a vector with elements ranging over the interval $[0, 2\pi]$. Polar profiles for different light sources are shown in Fig. 3.1.

As a reminder, polar profiles map each angle φ a magnitude r forming a function $r(\phi)$. They are circular axis versions of the domain $[0..2\pi]$, with negative magnitudes point reflected by 180 degrees. Thus, they are functions with an angular domain making them directionally dependent, which is a well suited representation for point lights (the type of light source with theoretically no finite extend), as we use them in a style of Dachsbacher et al. [35]. For the sake of interpretability, we exclude negative values (energies) from the scope of our calculations which we express through mathematical clip functions, limiting negative values to 0.0, which we denote by subscript $+$ in the following. We will describe the emission characteristics of point lights exchanging energies in the grid through such polar functions. Our grid consists of $dim = width \cdot height$ virtual point lights (VPLs). As a start-up thinker for the propagation scheme, we will think of a grid cell in 2D as a center pixel within an 8-neighborhood with a previously injected point light (imagine a spherical emitter $r = 1$ here for simplicity). Inside of this cell, we imagine a circular polar profile of radiant intensities for every angle ω given in $[\frac{W}{rad}]$. Remember, that all considerations from now on are given concerning a single grid cell respectively. Each cell has 8 (4 faces/edges and 4 diagonals) unique neighbors, which are adjacent point lights in the grid. We consider one face neighbor (top) to shortly explain the steps of the scheme. The diagonals and the others are obtained from derivation and

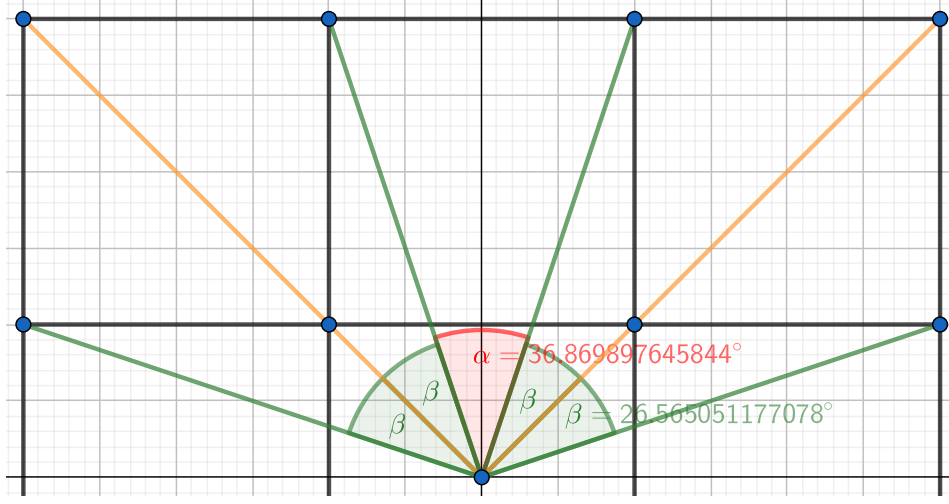


Figure 3.2: Propagation Scheme

symmetry. Naively, one would consider to just transmit the intensity in the direction its pointing into. But this would omit every propagation attenuation principles (any principles at all specifically), if one takes a closer look.

Accumulation stage Each cone type (yellow and green) is an individual neighbor-dependent angular area. The diagonals take in a greater angular domain while they are located at greater distance and thus need to be attenuated more strongly (consider an isotropic stimulus). We aim for distributing contributions for overlapping cones to transmit attenuated (because of the larger factor $\sqrt{2}$ distance) intensity in form of a shared part to the diagonals in a single step (which also prevents from zigzagging diagonal intensities in case we would transmit diagonal intensities around the corner). Therefore, the intensity gathered in polar coordinates in areas marked in green, needs to be partitioned between 2 of the 8 cone index neighbors with following considerations:

The face neighbors take in the angular domain of $d_1 = \alpha + 2\beta$ (yellow cone), whereas the diagonal neighbors (green cones) span a domain of $d_2 = 2\beta$. The linear combination weights for the overlapping (green) parts (β) are obtained from the relative overlapping angular area of the angular span β w.r.t. diagonal cones (2β) and yellow face cones ($\alpha + 2\beta = 90^\circ$) (yellow cone is overlapped by green cones at full angle). The linear combination weights are subsequently normalized to $\sum = 1.0$ (normalization condition):

$$\varepsilon_1 = \frac{\frac{\beta}{\alpha+2\beta}}{\frac{\beta}{\alpha+2\beta} + \frac{\beta}{2\beta}} \approx 0.362291 \quad (3.1)$$

$$\varepsilon_2 = \frac{\frac{\beta}{2\beta}}{\frac{\beta}{\alpha+2\beta} + \frac{\beta}{2\beta}} \approx 0.63771. \quad (3.2)$$

Remark, that we can precompute these linear combination weights and reutilize them in each iteration. We define the energetic contributions as follows:

$$\Phi_\alpha = \varepsilon_1 \int_\alpha I(\omega) d\omega = \varepsilon_1 \left(\int_{-\pi/4}^{-\alpha/2} I(\omega) d\omega + \int_{\alpha/2}^{\pi/4} I(\omega) d\omega \right) + \int_{-\alpha/2}^{\alpha/2} I(\omega) d\omega \quad (3.3)$$

$$\Phi_\beta = \varepsilon_1 \int_\beta I(\omega) d\omega = \varepsilon_2 \int_{-\beta}^{\beta} I(\omega) d\omega \quad (3.4)$$

Injection stage We place a cosine lobe scaled with the integrated (accumulated) energies in each of the 8 cone directions (angular domains), whereas outgoing integrated and summed energies are weighted by the linear combination weights (factors) in overlapping (green) areas to split up the contributions accordingly. The contribution areas are depicted in Fig. 3.2, the cosine lobes can be observed in Fig. 3.3 respectively. This manner of propagation is inspired by the propagation scheme of Dachsbacher et al. [35] (see the paper on how to “project the flux into a point light”). Basically, we sum up the weighted intensities corresponding to the respective neighbor index and scale an oriented cosine lobe (in direction of the neighbor) with the total flux Φ_t accumulated inside the neighbor range to transmit all of the parsed intensity for each neighbor distributed into a cosine distribution. We repeat this process once for each neighbor and each cell respectively. When we are done with the whole grid, we just recursively use the result and keep on running propagation cycles. The implementation is done in a dual buffer approach which pushes the energies back and forth (buffer A to buffer B) for each propagation cycle until a state of convergence is reached, which is the case when the energy is spreaded throughout the grid and enters a stationary state of equilibrium, characterized by equal out (at grid borders) to in (at light source positions) energy-flow with no more nettings going on inside the domain. We measure the convergence by setting up an overall distribution error w.r.t to the previous iteration. Note that even though we propagate the energy directly through the diagonal cones (which forms a square initially), the 8 degrees of freedom introduce a circular (spherical) wave front after few iterations (for greater resolutions) as depicted in Fig. 3.3. This occurs, because of a faster propagation attenuation which counteracts the larger angle of diagonal cones. There is a redistribution of intensities in the long term resulting in a, since the shared part effects a uniformly spaced isotropic propagation profile resulting in a circular torus-shaped distribution of intensities, as also observed in Fig. 3.3. Also note, that this approach satisfies energy conservation and propagation attenuation principles and applies to the principles of light propagation in vacuum (by default). We infer, that we can assume to follow physical models of spherical waves. The verification of these principal requirements and specifications can be observed in the evaluation in chap. 4.

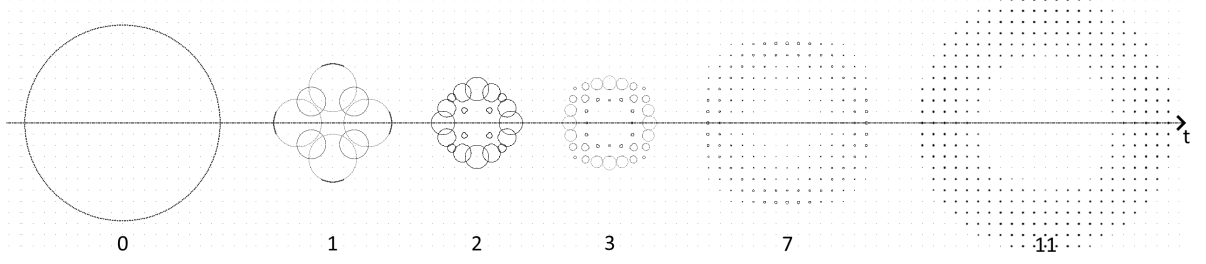


Figure 3.3: Impulse response for isotropic stimulus (few iterations: t)

We first show the approach for a few iterations with a manual stop criterion to have a kind of imagination for the propagation process:

Fig 3.3 depicts the named iteration indices 0,1,2,3,7,11 for the propagation scheme for a circular pulse stimulus which gets turned off after 1 iteration to demonstrate raw energy spreading in the field (impulse response of the algorithm). In step 1 one can observe the placing of cosine lobes scaled with the respective accumulated and weighted energy for the summed area in each of the 8 neighbor directions. In the subsequent steps, remark the evolution from an 8-neighborhood in step 1 to a circular shaped torus.

As already mentioned, a dual-buffer approach is chosen here, as we can efficiently propagate from one into the other and just swap pointers to start all over (after reinitialization) until convergence. Another reason is it yields a reasonable source-target or read-write segmentation. We introduce a new operator called *propagate*, permitting us to pre-define the propagation in a symbolic manner. The propagation definition denotes the transformation from one grid (A-buffer) into another (B-buffer) through proper application of the *propagate*-operator to each cell. We will extend this operator with a normalization correction in sect. 3.3 in form of a precise mathematical formulation respectively. For now, just imagine the approach in raw operating mode, with no tensor influence incorporated. That is, evaluating and accumulating the polar profiles, weighting with the linear combination weights and scaling and placing of a cosine lobe to the respective neighbor. For now, we want to define our operator *propagate*: Each Grid or Buffer is basically a set of cells defined as $\mathcal{C} = \{\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_i\}$. The dual buffer result \mathcal{C}' is denoted symbolically as follows for all cells with a mean intensity greater than zero (which do not contain trivial null samples):

$$\mathcal{C}' = \text{propagate}\{\mathbf{c}_i \in \mathcal{C} \mid |\bar{I}_i| > 0\} \quad \forall i \in [1, \dim]. \quad (3.5)$$

where: i : cell index, I : source buffer radiant intensity

The dual buffer result \mathcal{C}' denotes the set of all transformed cells after the propagation,

i.e., after applying the operator *propagate* to each cell. This result is generated once in each iteration and gets pushed back and forth between the 2 buffers. Note that, $dim = width \cdot height$, which equals the tensor field resolution in this case. At last, we define a threshold for the minimum overall distribution error for the total radiant flux taken into account for a convergence criterion, initiating a stop sequence when falling below that threshold ϵ (, e.g., 0.5) :

Criterion:

$$\Delta\Phi_{total} = \sum_{c_i \in \mathcal{C}} abs \Delta I_i(\omega) = \sum_{c_i \in \mathcal{C}} \int_0^{2\pi} abs(I'_i(\omega) - I_i(\omega)) d\omega \stackrel{!}{<} \epsilon$$

where ϕ_{total} : total radiant flux, I' : target buffer radiant intensity

Note that we take the absolute value of the differences to prevent from mutual compensation.

Until now, we aimed to simulate the propagation of light in empty space (vacuum) providing us with a physically-motivated but very simplified base-approach. We aim to modulate the transmission of the intensities on top of the base approach with transmission profiles obtained from the tensor fields through principal component analysis, which yields ellipsoid glyph equations in the following.

3.3 Transmission Profiles and Weighting Functions

Note that the following considerations are only given for the tensor profile operating mode with tensor influence incorporated. To obtain a kind of footprint of the tensor field, we set up ellipse equations for each glyph representation, obtained from applying PCA on each cell (tensor) in the grid. Concurrently, we extend our abstract mathematical operator *propagate*, previously referenced in sect 3.5, by a normalization strategy necessary for windowing weighting profiles, as it is the case for our elliptical transmission profiles, which we will capture in the following. At this point, we presume the precomputation of singular value decomposition and linear combination weights. We map the singular values (corresponding to eigenvalues λ here for simplicity) in decreasing order to the ellipses radii (half-axes) $a(\lambda_1)$ and $b(\lambda_2)$ as follows:

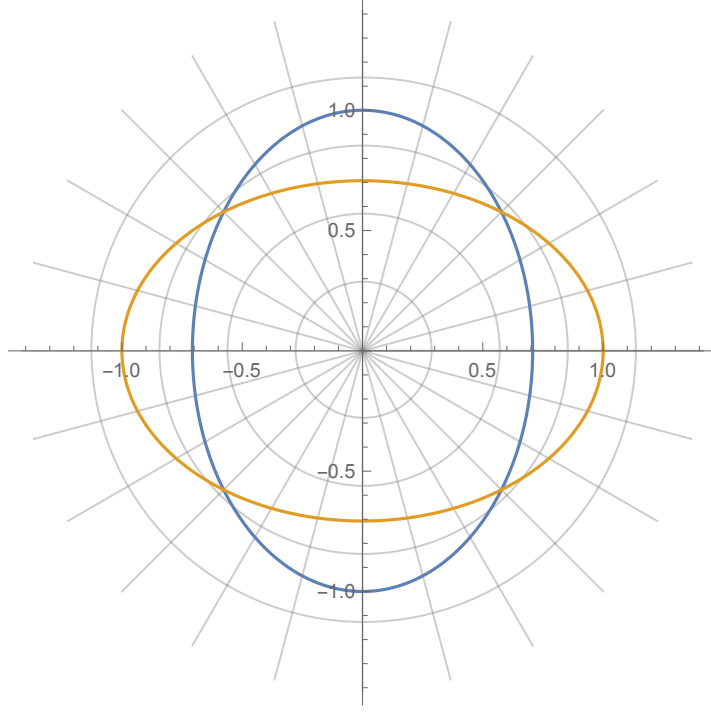


Figure 3.4: red: Intensity Profile, cyan: Transmission Profile (glyph Ellipsoid Eq.)

Ellipse Equation:

$$r(\omega) = \frac{ab}{\sqrt{a^2 \sin^2(\omega - \varphi) + b^2 \cos^2(\omega - \varphi)}} = \frac{\lambda_1 \lambda_2}{\sqrt{\lambda_1^2 \sin^2(\omega - \varphi) + \lambda_2^2 \cos^2(\omega - \varphi)}}. \quad (3.6)$$

That is, the singular values form the half-axes of the PCA ellipsoid, which yields us a symbolic definition for our transmission (transfer) function as weighting profiles which we can now evaluate as a parsing and accumulating step. We compute the offset angle φ w.r.t. the x-axis by exploiting the *atan2*-function with 4-quadrant evaluation for the first singular vector (ordered decreasingly in value) because we scale the x -axis with the first singular value. As a preliminary preparation, this profile is precomputed (sampled) for all discrete angular steps for each cell (, e.g., 360) and stored for reutilization as it is the case for the previously mentioned linear combination weights as well. The weighting scheme which we apply is depicted in Fig. 3.4. We interpret the polar functions as discrete, sampled 1D vectors with a uniform angular resolution. That means, we can use element-wise multiplication for the 1D-vectors in red (Intensity Profile) and cyan (Transmission Profile) to obtain a windowed version of the intensity profile and perform the following integration step to accumulate the total radiant power (flux) as follows. Note that we model angular dependent radiant intensity $I(\omega)[\frac{W}{rad}]$ as an underlying physical entity, which is well suited for describing point light sources. As a side note,

3 Method

the physical quantity perceived as brightness by human eyes, is the radiance (luminance), which is yielded by relating radiant intensity to a finite sensitive area ($[\frac{W}{rad}/m^2]$). The primary definition for our total accumulated incident radiant flux (presuming that the cone is given in local coordinates - i.e. centered around the origin) is then denoted as:

$$\Phi_\alpha = \varepsilon_1 \left(\int_{-\pi/4}^{-\alpha/2} T(\omega) I(\omega) d\omega + \int_{\alpha/2}^{\pi/4} T(\omega) I(\omega) d\omega \right) + \int_{-\alpha/2}^{\alpha/2} T(\omega) I(\omega) d\omega \quad (3.7)$$

$$\Phi_\beta = \varepsilon_2 \int_{-\beta}^{\beta} T(\omega) I(\omega) d\omega \quad (3.8)$$

This permits us to scale an oriented and clipped cosine-lobe \cos_+ with the total radiant flux w. respect to its own total energy when integrated likewise. A discrete cosine lobe never reaches its theoretical sum of 2 exactly in the interval $[-\frac{\pi}{2}, \frac{\pi}{2}]$. Therefore it needs to be computed to account for floating-point precision errors and normalized by following relation (, e.g., $\sum_{\cos} \approx 1.9949204635834517$).

$$\sum_{\cos_+} = \int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} \cos_+(\omega) d\omega \approx 2, \quad (3.9)$$

$$\cos_n(\omega) = \frac{\Phi_t}{\sum_{\cos_+}} \cos(\omega). \quad (3.10)$$

This yields a cosine lobe, which has an overall energy of Φ_t if the point light is interpreted with a Lambert-emitter (cf. [?]) characteristic (as part or area element of a diffusely reflecting surface) since the intensity profile matches a cosine in this case. This emitter characteristic is the case for global illumination diffuse indirect lighting (non-specular). It is used in similar style by Dachsbacher et al. for propagating intensities through LPVs (light propagation volumes). But hey we should not generate any exceeding energy right? Now each tensor ellipse equation has a mean transmission rate [%], which turns out to be a measure or synonym for the total rate of the outgoing transmitted radiant intensity $\int_0^{2\pi} T(\omega) d\omega$. Since energy can not be generated from transmission profiles (only transmitting, not generating), we have to normalize every tensor ellipse equation with the highest mean transmission rate in the field, which is then transmitting the whole outgoing intensity $\int_0^{2\pi} I(\omega) d\omega$ in total. This restricts the tensor ellipse mean to a maximum of 100% for the tensor with the highest tensor magnitude. We will use the self-made definition of mean tensor magnitude \bar{r}_{max} obtained from the PCA here for simplicity. Hence the tensor ellipse equations are normalized through the following

step:

$$r_n(\omega) = \frac{1}{\bar{r}_{max}} r(\omega). \quad (3.11)$$

Notice, that this implements the concept of absorption implicitly, because tensors with a lower mean \bar{r} than 100% lose energy whereas absorption equals $\mu = 1 - \bar{r}$ here. Now that we have coped with the problem of overall transmission, we are left with another problem that occurs with the weighting functions as they can still yield values above 1.0. Consider following examples as $T(\omega)I(\omega)$ pairs: $1.5 \cdot 1.5 = 2.25$ and $0.5 \cdot 0.5 = 0.25$, which disproves us that a mean $\bar{r} = 1.0$ necessarily effects a mean transmission intensity I_t of $1.0 < 1.25 = \frac{2.5}{2}$. This would potentially generate intensity out of nothing which leads to crucially unstable behaviour. Well, but only in case energy conservation principles were not respected. To construct a solution, we ask for the source of the excessive energy. If the tensor mean complies with the constraint $\bar{r} \leq 1.0$, then the energy must lack in some other place (since the overall transmission fulfills the energy conservation constraint in that case). The only problem is that a normalization condition in usual style for the transmission profile $T(\omega)$ does not necessarily imply the same normalization for the term $T(\omega)I(\omega)$. However, a proper normalization can provide corrective support, which reveals another subsequent normalization constraint. The normalization factor is defined in 3 subsequent steps:

1. Normalization of TI to $\overline{TI} = 1.0$
2. Subsequent scaling w. mean intensity \bar{I} for energy conservation principles
3. Subsequent scaling w. mean tensor magnitude \bar{T} for absorption principles

Hence the transmitted outgoing intensity for a single direction is given as:

$$I_t(\omega) = n_f \varepsilon_d T(\omega) I(\omega) \quad w. \quad n_f = \frac{\bar{T}\bar{I}}{\overline{TI}}. \quad (3.12)$$

This transforms our equation for the transmitted radiant flux into:

$$\Phi_{\alpha,t,i} = \int_{\alpha} I_{t,i}(\omega) d\omega = n_f \Phi_{\alpha,i}, \quad (3.13)$$

$$\Phi_{\beta,t,i} = \int_{\beta} I_{t,i}(\omega) d\omega = n_f \Phi_{\beta,i}. \quad (3.14)$$

Which is now respecting energy conservation principles very conveniently, as all we need is a set of means and we are done. This collapses our abstract mathematical operator *propagate*, which is applied to each cell, into the following steps:

1. Computation of normalization factor n_f
2. Direction (component)-wise weighting to account for shared part in diagonal cones
3. Integration (accumulation) total radiant flux weighted with normalized transmission profiles inside the angular neighbor range (read-access)
4. Scaling of cosine lobe corresponding to neighbor direction and subsequent placing to corresponding neighbor (write-access)

Remember that this operator needs to be applied once to each cell to form a whole propagation cycle propagating intensities from source to target buffer. This approach ensures that no energy will be generated and therefore implements the physical law of energy conservation or Kirchhoff's current law if you want to think of a particle's diffusion process. Consequently, this allows us to assume that all sources of power (energy) are user-defined. In fact the approach has proven to be robust w.r.t. energy losses for several iterations as depicted in chap. 4 in the stochastic error graph.

3.4 Physical Model

To pose a physical interpretation for the transmission of the tensor field, we imagine a melting crystal with fibrous structures aligning with the major eigenvectors of the tensor field. This is what we call the footprint of the tensor field. In that sense of imagination, we would virtually imprint every tensor field into a crystalline structure as a preliminary initialization step. For an every day example, this can be observed in, e.g., the gemstone tiger's eye cat's eye effect, where crystal fibers align with the crystal axes respectively. The probability density function $P(\omega)$, for a photon being scattered in a particular direction ω ??, is then described as:

$$P(\omega) = \frac{T(\omega)}{\int_{k\pi}^{(k+1)\pi} T(\omega) d\omega} \quad (3.15)$$

$$w. \int_0^{2\pi} P(\omega) = 1. \quad (3.16)$$

This probability density function (PDF) indicates whether there is directed anisotropy or spherical isotropy w.r.t the neighboring 180 (πrad) degrees and how likely the spontaneous scattering of a photon in this particular direction will happen. Each tensor is then a unique footprint which can be precomputed for convenience and described as polar profiles just as our previously mentioned intensity profiles. We apply these polar profiles in a sense of transmission profiles as discussed in sect. 3.3.

3.5 Light Transport Gradient (LTG)

Remark that we denote the final buffer result $\mathbf{b} = \mathcal{C}_{final}$ in the following:

Now, that we are able to propagate intensity in grid, directed by the eigenvectors of the tensor field, we aim to do something more elaborate to segment ROIs (regions of interest) characterised by divergence/convergence of tensor field lines and drastic changes in anisotropy. As tensor field lines are bidirectional, convergence and divergence can not reasonably discerned in that case. We aim to extract ridges from a FTLE-like field to grasp the key regions separating anisotropy behavior in the field segmented as height ridges in the field, which are in turn LCS (Lagrangian Coherent Structures) well known for their rich substance in physical meaning. They can even be observed in nature for, e.g., vortices in flow fields. For that, we will use our, previously defined, *propagate*-operator on every possible light source location \mathbf{x} and direction ω in the grid to obtain the final set of buffers $\mathcal{B} = \{\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_j\}$ of impulse responses of the algorithm:

$$\mathcal{B} = \text{propagate}\{c_i \in \mathcal{C} \mid |\bar{I}_i| > 0\} \forall i \in [1, \text{dim}] \forall j \in [1, \text{dim} \cdot \text{steps}]. \quad (3.17)$$

Whereas j is the index of light source position and direction representing the initialization in each iteration and *steps* is the count of the discrete angular steps. The initialization is done by placing a light src (delta-impulse) with $\bar{I}_i = 1.0$ at the exact location of index j realizing a light source at location $(x \mid y)$ in direction ω . Eq. 3.17 now denotes the set of buffers \mathcal{B} containing light distributions propagated from every possible direction and location. Now, that we have measured impulse responses from every location in our long 1D sample vector, we can incorporate finite (central) differences to obtain the gradient of the set of final buffers with final light distributions as elements $\mathcal{C}_{final} = \mathbf{b} \in \mathcal{B}$:

$$\nabla \mathbf{b}_{x,y,\omega} = \begin{pmatrix} \frac{\partial \mathbf{b}_{x,y,\omega}}{\partial x} \\ \frac{\partial \mathbf{b}_{x,y,\omega}}{\partial y} \\ \frac{\partial \mathbf{b}_{x,y,\omega}}{\partial \omega} \end{pmatrix} \approx \frac{1}{2} \begin{pmatrix} \sum_{\mathbf{c}} |\mathbf{b}_{x+1,y,\omega} - \mathbf{b}_{x-1,y,\omega}| \\ \sum_{\mathbf{c}} |\mathbf{b}_{x,y+1,\omega} - \mathbf{b}_{x,y-1,\omega}| \\ \sum_{\mathbf{c}} |\mathbf{b}_{x,y,\omega+1} - \mathbf{b}_{x,y,\omega-1}| \end{pmatrix} = \frac{1}{2} \begin{pmatrix} \sum_{\mathbf{c} \in \mathbf{b}_{\mathbf{x}}} |\mathbf{b}_{\mathbf{x}}| \\ \sum_{\mathbf{c} \in \mathbf{b}_{\mathbf{y}}} |\mathbf{b}_{\mathbf{y}}| \\ \sum_{\mathbf{c} \in \mathbf{b}_{\omega}} |\mathbf{b}_{\omega}| \end{pmatrix}. \quad (3.18)$$

Note that this step is very expensive considering runtime-costs since it follows the following runtime observation. We have $\text{width} \cdot \text{height} \cdot \text{steps}$ numbers in the grid and on top $\text{width} \cdot \text{height} \cdot \text{steps}$ buffers in total, which need to be processed. Also we remark the convergence criterion which introduces another factor width and set $\text{height} = \text{width}$ here for simplicity, which yields:

$$\mathcal{O}(\text{width width steps width width steps}) = \mathcal{O}(\text{width}^5 \text{ steps}^2). \quad (3.19)$$

3 Method

, which is the runtime behavior in order notation considering memory consumption and iteration step count. It is noticeable that the runtime increases fast with increasing resolution, which makes parallelization algorithms a crucial tool to compute the results in reasonable time with additional computing power provided by a computer cluster. Next, we compute the Euclidean norm of the Global Illumination Gradient yielding a scalar field, which can be visualized directly or via extracted ridges in a visualization framework like ParaView from VTK format:

$$|\nabla \mathbf{b}_{x,y,\omega}| = \sqrt{\left(\frac{\partial \mathbf{b}_{x,y,\omega}}{\partial x}\right)^2 + \left(\frac{\partial \mathbf{b}_{x,y,\omega}}{\partial y}\right)^2 + \left(\frac{\partial \mathbf{b}_{x,y,\omega}}{\partial \omega}\right)^2}. \quad (3.20)$$

Note that the resulting scalar field is $(n + 1)$ -dimensional (x, y and direction θ) and needs to be flattened by averaging or projection in 3D for proper visualization in case the approach will be adopted in 3D.

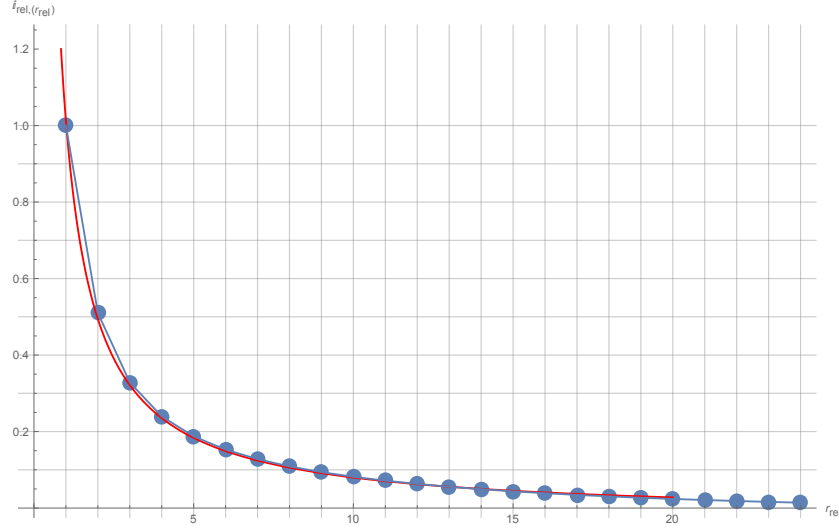


Figure 4.1: Propagation Attenuation

4 Results and Evaluation

4.1 Propagation Attenuation

It can be proven that the approach follows propagation attenuation principles and follows the inverse (square in $3D$) law ($\sim \frac{1}{r}$) as observed in the graph in Fig. 4.1. For that, we plot the relative intensity $i_{rel}(r_{rel})$ (in relation to the intensity at position (radius) 1 to account for the inverse square law, which holds for relative distances. We evaluate the relative intensity by forming a ratio $\frac{I(r)}{I_1}$, that relates the step intensity to the src intensity at position 1.

4.2 Light Propagation Scheme Behaviour

To evaluate the plausibility of the light distribution results obtained from the light propagation scheme, we plot the transmission profiles overlaid with the resulting intensity profiles for several synthetic test fields, whereas red: tensor glyphs (transmission profiles) and blue: intensity profiles are both depicted in polar coordinates: One can observe here, how light is propagated with no directional bias leading to an absolutely

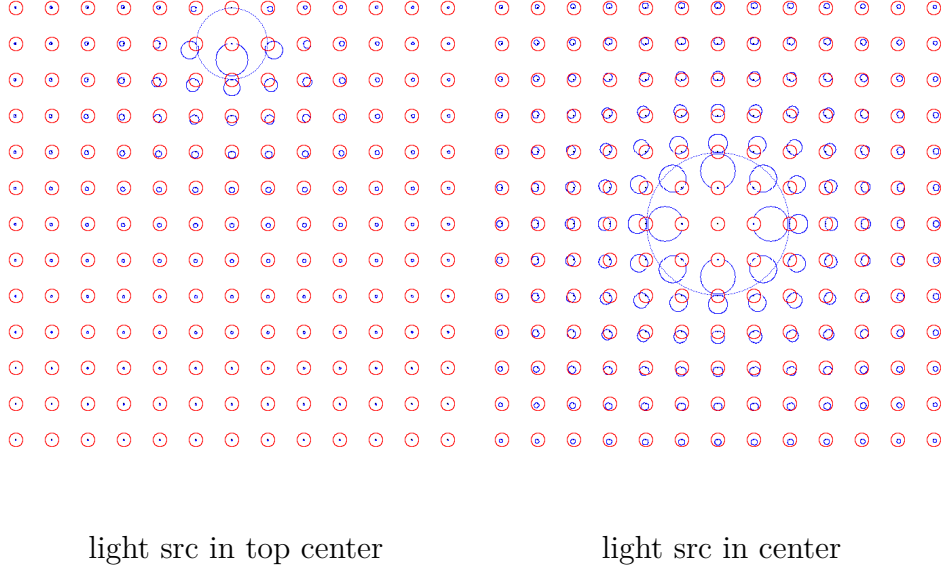


Figure 4.2: final light propagation distributions for “isotropy”-testfield

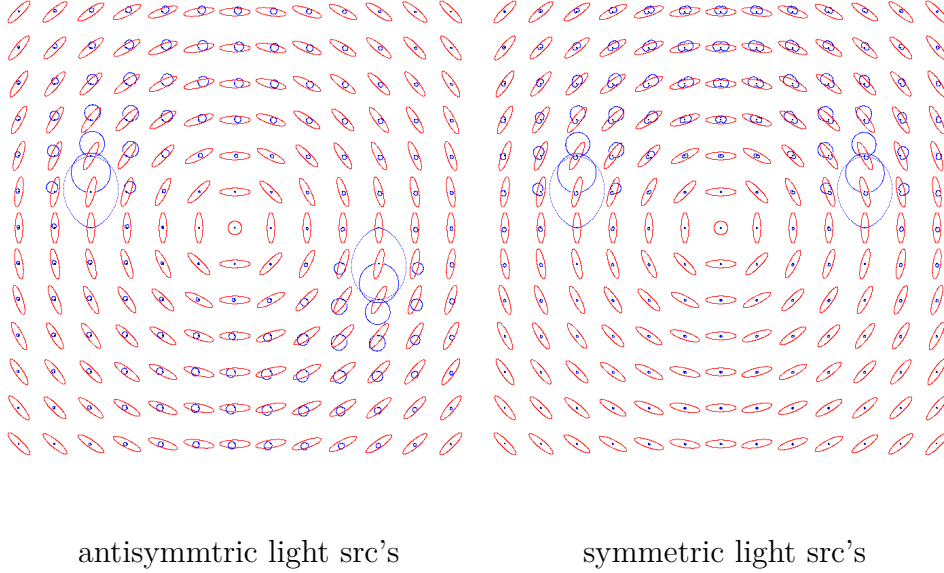


Figure 4.3: final light propagation distributions for “rings”-testfield

isotropic distribution in case b), which is symmetric as expected. This result is considered to be the absolute ground truth example for the light propagation scheme, which is evaluated separately in sect. 4.1 and 4.4. In this next example, we use the tensor field “rings” as input data for the approach and observe how the intensities are channeled in a circular orbit around the central origin. The anisotropy measures decide whether the transmission profile is more narrow (radical) or more wide (liberal). The results for our tensor field “spiral” can be interpreted in analogous fashion as the previously discussed “rings”-testfield. The influence incorporated by the tensor field is easily noticeable in

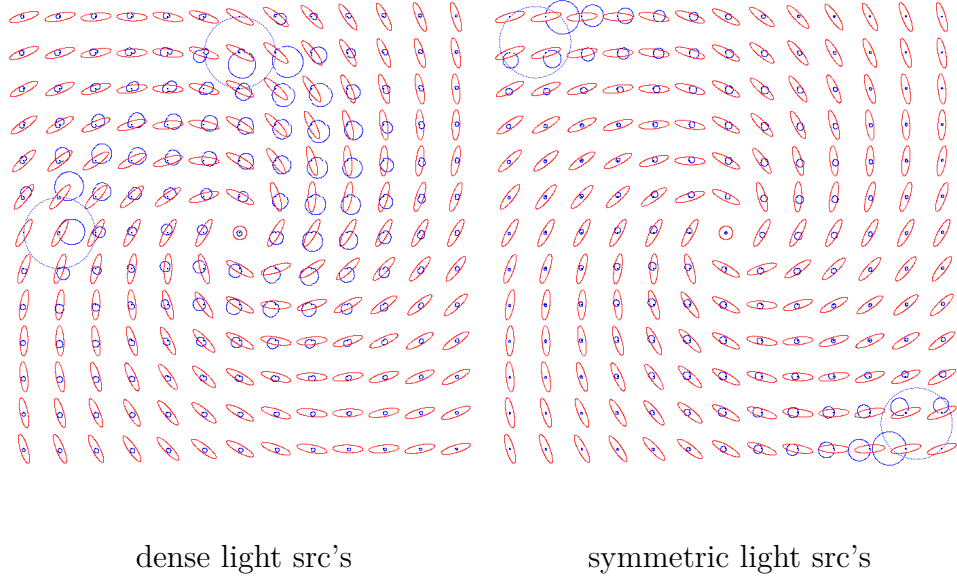


Figure 4.4: final light propagation distributions for “spiral”-testfield

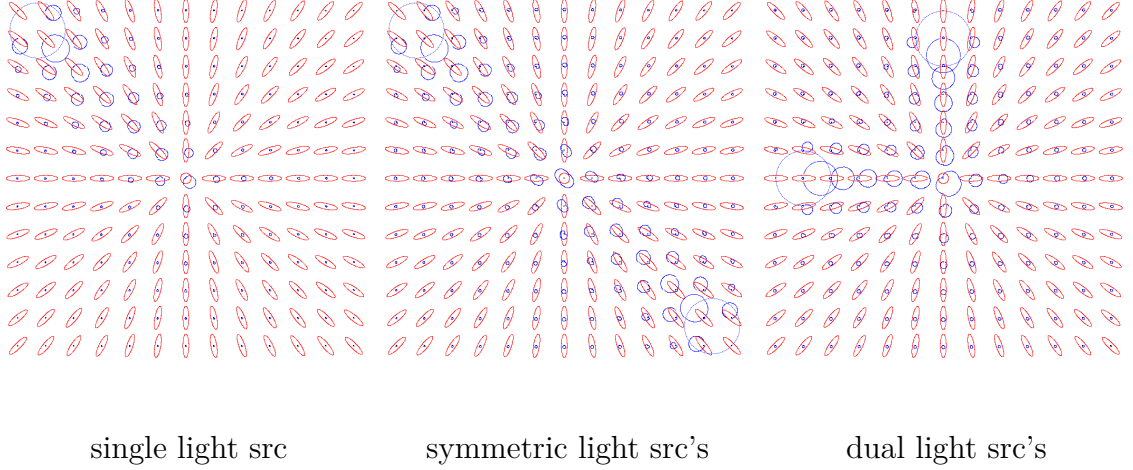


Figure 4.5: final light propagation distributions for “star”-testfield

this case as well.

The star example reveals another interesting feature of the approach: if the intensity impulsively hits an isotropic region, it gets split up into a range of the neighboring 180 degrees of directions.

Note, that a logarithmic scale needs to be incorporated to produce results on a large scale in a sense of higher spatial resolution (e.g. 100×100), since the propagation attenuation follows the distance to the origin inversely proportional.

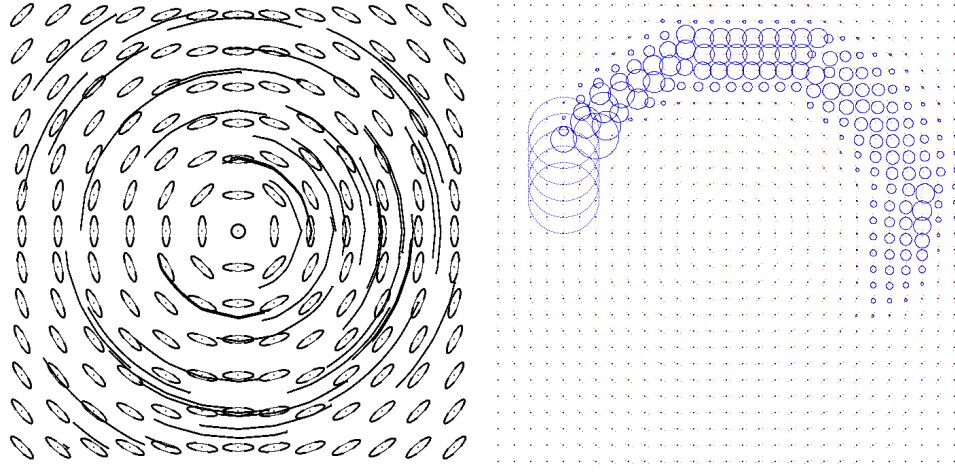


Figure 4.6: a) tensor field Lines and glyphs for “rings”-Testfield, b) intensity propagation in grid

4.3 Behaviour for Total Anisotropy

We also wanted to check for the limitations of the approach by having total linear anisotropy (the case whereas one eigenvalue $\lambda_1 > 0$ and the other $\lambda_2 = 0$) incorporated. For this we set up bidirectional delta-pulses as transmission profiles to have the whole energy transmitted in just one, discrete direction. What we can observe here, is kind of a sampling drift as the approach uses 8 discrete directions to propagate averaged intensity whereas the circle uses much more (namely infinite if presumed as perfect circle). Hence, the energy can never follow a perfect circle, but at best an 8-edged octagon. This means, that there is a drift of at least two cells per diagonal (one overshoot per enter and escape each) as observable in Fig. ???. Nonetheless, the energies are transmitted in circular orbits around the circle center.

4.4 Energy Conservation Test

The Energy Conservation Test was done in similar manner as the Propagation Attenuation Test in sect. 4.1. We measure the impulse response of the algorithm for about 80 iterations by summing the total energy in the grid after each iteration and comparing it to the initial energy sum in iteration 0 in absolute values. This approach yields the following graph:

The approach remains stable until iteration 40, which is at the time (point) when the numbers are too small and get rounded down under the double machine epsilon. You can see heavy fluctuations because the numbers get rounded upwards likewise. But what

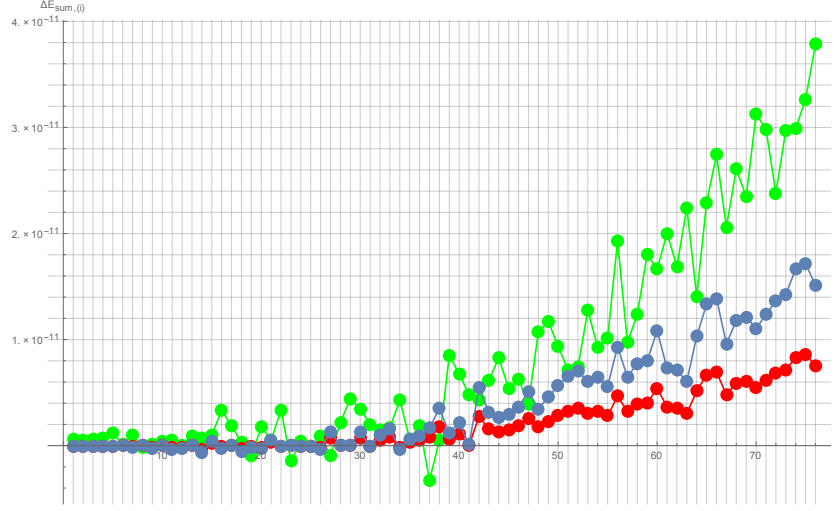


Figure 4.7: Sum Error: src intensities: red: 1.0, blue 2.0, green: 5,0

is lost, is lost and can not be gained back. The error is accumulating gradually and with faster slope for higher source intensities, since floating point precision, i.e., rounding errors operate more heavily on large numbers since it is a constant relative loss leading to a greater absolute ΔE .

4.5 Light transport gradient (LTG)

The light transport gradient approach, which is closely related to the FTLE approach in vector fields, is evaluated from simple ground truth examples to elaborate real datasets. As a little warmup, we will have an experiment exhibiting bare, raw tensor field lines diverging w.r.t. one axis.

What we can observe here, is called a “ridge” occurring as a high value line segment (red vertical track) which is already quite obvious for human vision. A gradient filter could be applied to extract and emphasize edges from the image as a preprocessing step for computer vision. These ridges represent ROIs with underlying Lagrangian coherent structures, which we already discussed in sect. 1.1. It could for example represent a stress field inside of a mechanical tool with a central axis with maximum stress condensation.

We also evaluated the approach for the “inverse” test field, whereas we can observe height ridges for the 90° slice in any position with diverging tensor field lines w.r.t. the light source direction. This is certainly not the case for the upper (missing) ridge whereas tensor field lines converge with respect to the UP (90°) direction. This proves that the method is capable of detecting ridges orthogonal to the light source direction, but not in antiparallel directions, which exhibit converging tensor field lines as bidirec-

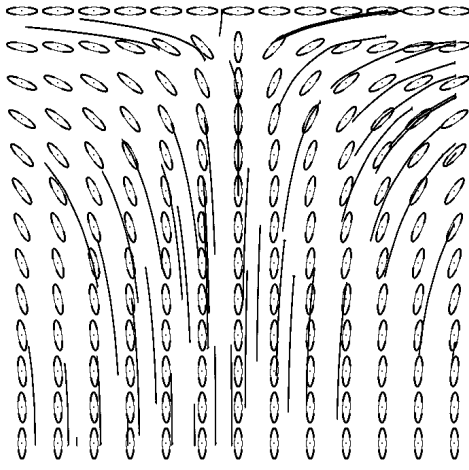


Figure 4.8: a) TFLs and glyphs for “drain” test field

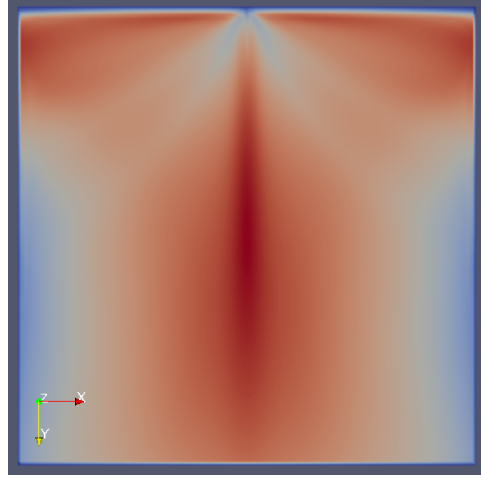


Figure 4.9: b) LTG(90°) for a)

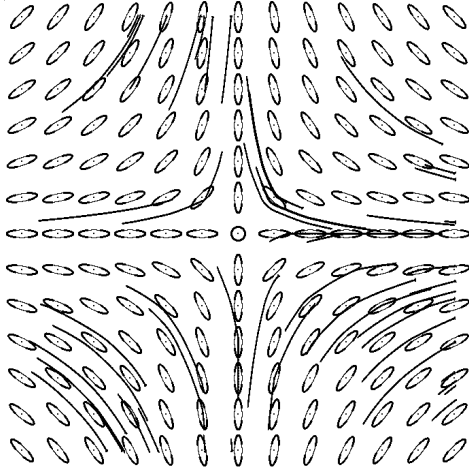


Figure 4.10: a) TFLs and glyphs for “inverse”-test field

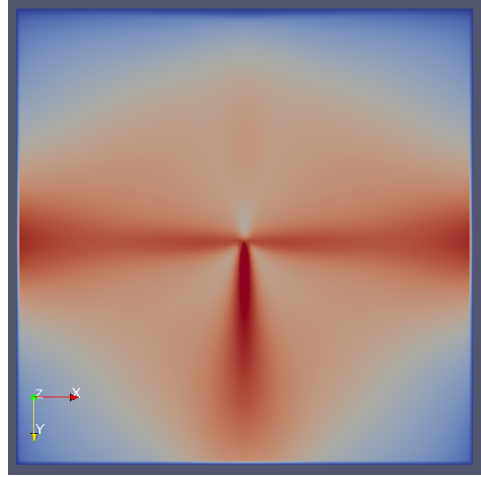


Figure 4.11: b) LTG(90°) for a)

tional counterpart. Luckily, we measured the gradient for every possible sampled light source direction which complements this information into another slice (and the resulting volume of stacked slices). As we are also interested in the whole ensemble of directions, we can use a volume or surface representation for the $n + 1 = 3$ -dimensional grid which results from our FTLE computation. As a subsequent step, we segment height ridges as thresholded volumes from the whole volume grid. The volume grid is what we call the final footprint of the tensor field which is necessarily derived from the initial glyph footprint. From it, we can observe the behavior of the tensor field for any direction. We

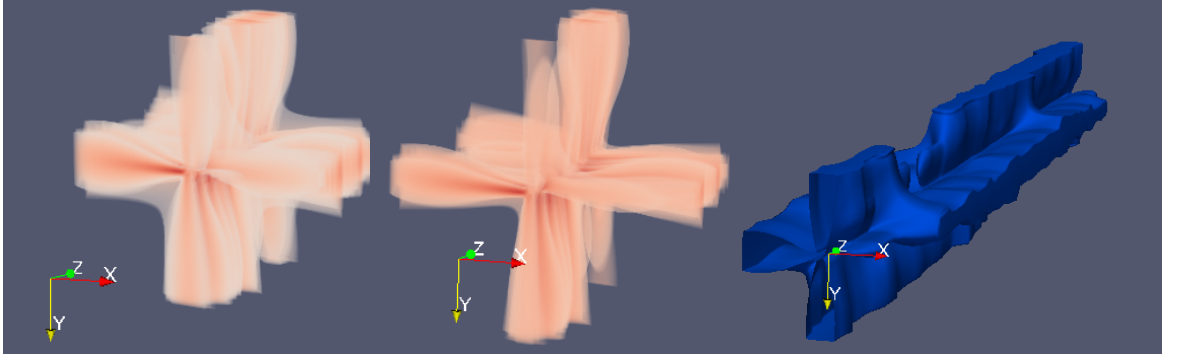


Figure 4.12: IsoVolume Filter on LTG volume representation: “inverse” test field

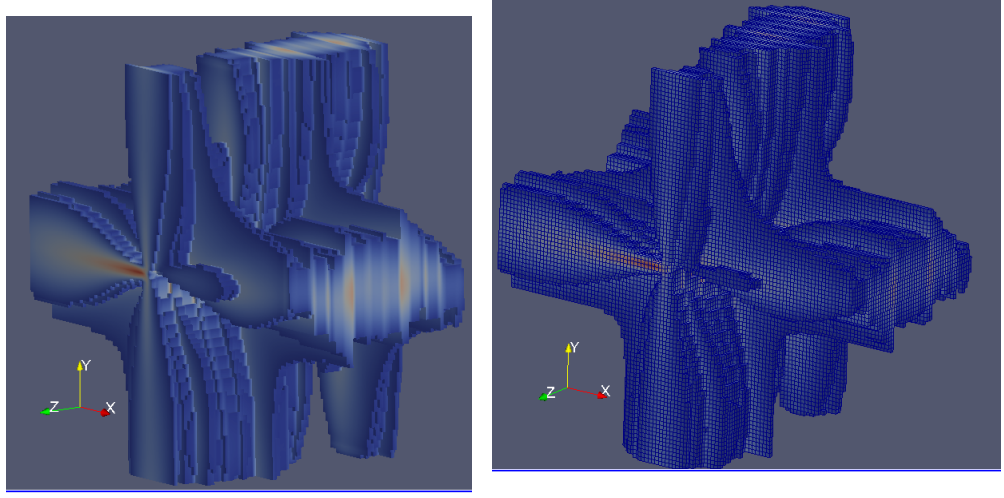


Figure 4.13: Contour filter on LTG stacked slice surface representations: “inverse” test field

consider extracted subsets of the volume grid as 3D height ridges in the domain of ω . That is, the slices we observed are stacked along the z -axis in the following:

Note that the z -axis represents the light source direction angle ω in this case. The contour filter reveals the star-shaped structure of the tensor field which rotates along the angle axis. This is the effect we already captured for the single slice: For each axis direction we can detect ridges in parallel and orthogonal to our predefined stimulus (light source) direction, but not in antiparallel direction. This causes the star shape to miss one branch or ray which is in turn the antiparallel direction missing for each light source direction emphasizing the impression of rotation. We can also try to segment height ridges represented as surfaces from the resulting volume grids as depicted in Fig 4.14

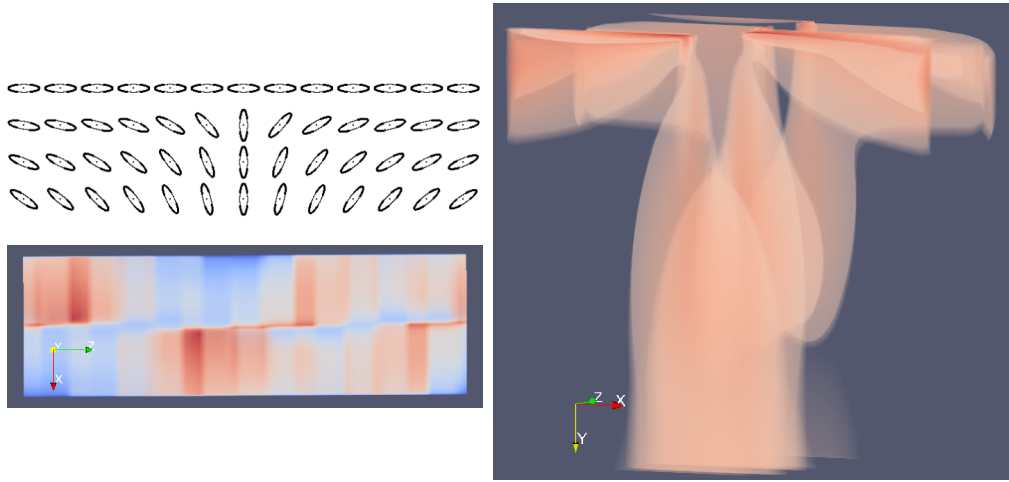


Figure 4.14: Contour filter on LTG stacked slice surface representations: “inverse” test field

5 Zusammenfassung und Ausblick

Hier werden noch einmal die wichtigsten Ergebnisse und Erkenntnisse der Arbeit zusammengefasst (nicht einfach eine Wiederholung des Aufbaus der vorherigen Kapitel!), welche neuen Konzepte, Methoden und Werkzeuge Neues entwickelt wurden, welche Probleme nun (effizienter) gelöst werden können, und es wird ein Ausblick auf weiterführende Arbeiten gegeben (z.B. was Sie machen würden, wenn Sie noch 6 Monate mehr Zeit hätten).

Bibliography

- [1] X. Zheng and A. Pang, “2d asymmetric tensor analysis,” pp. 3–10, 11 2005.
- [2] Z. Lin, H. Yeh, R. Laramée, and E. Zhang, “2d asymmetric tensor field topology,” 11 2012.
- [3] C. Moler, “Numerical Computing With MATLAB, 2004 / Bug Report Revives Interest in SVD Option of ?Eigshow?.” <https://blogs.mathworks.com/cleve/2016/08/08/bug-report-revives-interest-in-svd-option-of-eigshow/>, 2016. [Online; accessed 22-05-19].
- [4] M. Kieburg, “What is the Relation between Eigenvalues & Singular Values?.” http://wt.iam.uni-bonn.de/fileadmin/WT/Inhalt/people/Patrik_Ferrari/Conference2016/TalkKieburg.pdf, 2016. [Online; accessed 22-05-19].
- [5] L. Mroz, “What is the Relation between Eigenvalues & Singular Values?.” <https://www.cg.tuwien.ac.at/research/vis/seminar9596/2-topo/evinter.html>, 2016. [Online; accessed 22-05-19].
- [6] M. Hlawatsch, J. E. Vollrath, F. Sadlo, and D. Weiskopf, “Coherent structures of characteristic curves in symmetric second order tensor fields,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, pp. 781–794, June 2011.
- [7] T. Zirr, M. Ament, and C. Dachsbacher, “Visualization of coherent structures of light transport,” *Comput. Graph. Forum*, vol. 34, pp. 491–500, June 2015.
- [8] A. Vilanova, G. Berenschot, and C. van Pul, “Dti visualization with streamsurfaces and evenly-spaced volume seeding,” in *Proceedings of the Sixth Joint Eurographics - IEEE TCVG Conference on Visualization, VISSYM’04*, (Aire-la-Ville, Switzerland, Switzerland), pp. 173–182, Eurographics Association, 2004.
- [9] D. Weinstein, G. Kindlmann, and E. Lundberg, “Tensorlines: Advection-diffusion based propagation through diffusion tensor fields,” in *Proceedings of the Conference on Visualization ’99: Celebrating Ten Years, VIS ’99*, (Los Alamitos, CA, USA), pp. 249–253, IEEE Computer Society Press, 1999.

- [10] T. Delmarcelle and L. Hesselink, “Visualizing second order tensor fields with hyperstreamlines,” 03 1995.
- [11] X. Zheng, B. Parlett, and A. Pang, “Topological structures of 3d tensor fields,” pp. 551– 558, 11 2005.
- [12] B. Spencer, R. S. Laramée, G. Chen, and E. Zhang, “Evenly spaced streamlines for surfaces: An image-based approach,” *Comput. Graph. Forum*, vol. 28, pp. 1618–1631, 2009.
- [13] T. McLoughlin, R. S. Laramée, R. Peikert, F. H. Post, and M. Chen, “Over two decades of integration-based, geometric flow visualization,” *Computer Graphics Forum*, vol. 29, no. 6, pp. 1807–1829, 2010.
- [14] L. Feng, I. Hotz, B. Hamann, and K. Joy, “Anisotropic noise samples,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, pp. 342–354, March 2008.
- [15] T. Schultz and G. L. Kindlmann, “Superquadric glyphs for symmetric second-order tensors,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 16, pp. 1595–1604, Nov. 2010.
- [16] W. C. de Leeuw and J. J. van Wijk, “A probe for local flow field visualization,” in *Proceedings Visualization '93*, pp. 39–45, Oct 1993.
- [17] N. M. A. Worth, “Fusion of MRI data for visualization of white matter bundles.” <http://neuromorphometrics.org/papers/Worth/WorthMICCAI98reject.pdf>, 2016. [Online; accessed 22-05-19].
- [18] C. Pierpaoli and P. J. Basser, “Toward a quantitative assessment of diffusion anisotropy,” *Magnetic resonance in medicine*, vol. 36 6, pp. 893–906, 1996.
- [19] C.-F. Westin, S. E. Maier, H. Mamata, A. Nabavi, F. A. Jolesz, and R. Kikinis, “Processing and visualization for diffusion tensor mri,” *Medical image analysis*, vol. 6, no. 2, pp. 93–108, 2002.
- [20] G. Kindlmann, “Superquadric tensor glyphs,” in *Proceedings of the Sixth Joint Eurographics - IEEE TCVG Conference on Visualization, VISSYM'04*, (Aire-la-Ville, Switzerland, Switzerland), pp. 147–154, Eurographics Association, 2004.
- [21] P. Basser, “Microstructural and physiological features of tissues elucidated by quantitative-dicusion-tensor mri,” *J Magn Reson*, vol. 213, pp. 560–570, 01 2011.

- [22] C.-F. Westin, S. Peled, H. Gudbjartsson, R. Kikinis, and F. A. Jolesz, “Geometrical diffusion measures for MRI from tensor basis analysis,” in *ISMRM '97*, (Vancouver Canada), p. 1742, April 1997.
- [23] G. Kindlmann, D. Weinstein, and D. Hart, “Strategies for direct volume rendering of diffusion tensor fields,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 6, pp. 124–138, April 2000.
- [24] D. Le Bihan, “Looking into the functional architecture of the brain with diffusion mri,” *Nat. Rev.*, vol. 4, 06 2003.
- [25] S. Wakana, H. Jiang, L. Nagae, P. van zijl, and S. Mori, “Radiology fiber tract ? based atlas of human white matter anatomy,” *Radiology*, vol. 230, pp. 77–87, 02 2004.
- [26] X. Tricoche, *Vector and Tensor Field Topology Simplification, Tracking and Visualization*. PhD thesis, 01 2002.
- [27] F. Zhang and E. R. Hancock, “Edge detection and anisotropic diffusion for tensor-valued images,” in *2006 International Conference on Image Processing*, pp. 229–232, Oct 2006.
- [28] G. Kindlmann, D. B Ennis, R. T Whitaker, and C.-F. Westin, “Diffusion tensor analysis with invariant gradients and rotation tangents,” *IEEE transactions on medical imaging*, vol. 26, pp. 1483–99, 12 2007.
- [29] L. O’Donnell, W. Eric L. Grimson, and C.-F. Westin, “Interface detection in diffusion tensor mri,” vol. 3216, pp. 360–367, 09 2004.
- [30] G. Kindlmann, X. Tricoche, and C.-F. Westin, “Anisotropy creases delineate white matter structure in diffusion tensor mri,” vol. 9, pp. 126–33, 02 2006.
- [31] D. Palke, Z. Lin, G. Chen, H. Yeh, P. Vincent, R. Laramée, and E. Zhang, “Asymmetric tensor field visualization for surfaces,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, pp. 1979–1988, Dec. 2011.
- [32] J. Palacios, H. Yeh, W. Wang, Y. Zhang, R. Laramée, R. Sharma, T. Schultz, and E. Zhang, “Feature surfaces in symmetric tensor fields based on eigenvalue manifold,” *IEEE transactions on visualization and computer graphics*, vol. 22, 10 2015.

Bibliography

- [33] G. Haller and G. Yuan, “Lagrangian coherent structures and mixing in two-dimensional turbulence,” *Phys. D*, vol. 147, pp. 352–370, Dec. 2000.
- [34] G. Haller, “Lagrangian coherent structures from approximate velocity data,” *Physics of Fluids*, vol. 14, no. 6, pp. 1851–1861, 2002.
- [35] A. Kaplanyan and C. Dachsbacher, “Cascaded light propagation volumes for real-time indirect illumination,” in *Proceedings of the 2010 ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, I3D ’10, (New York, NY, USA), pp. 99–107, ACM, 2010.